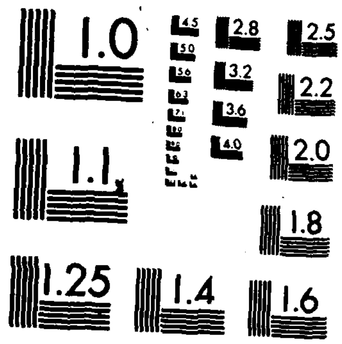
AD-A137 760 TIME DOMAIN ANALYSIS AND SYNTHESIS OF ROBUST CONTROLLERS FOR LARGE SCALE..(U) STEVENS INST OF TECH HOBOKEN N J DEPT OF MECHANICAL ENGINEERI.. 1/1

UNCLASSIFIED R K YEDAVALLI ET AL. 31 AUG 83 F/G 12/1 NL

END
FILMED
3-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A137760

(2)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE: -	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 84-0040
6a. NAME OF PERFORMING ORGANIZATION: Stevens Institute of Technology; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code): Department of Mechanical Engineering, Hoboken NJ 07030	7b. ADDRESS (City, State and ZIP Code): Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR; 8b. OFFICE SYMBOL (If applicable): NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: AFOSR-83-0139
8c. ADDRESS (City, State and ZIP Code): Bolling AFB DC 20332	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A6	

11. TITLE (Include Security Classification): SEE REMARKS ON BACK

12. PERSONAL AUTHOR(S): R.K. Yedavalli, R.N. Shanbhag and J. Irudayasamy

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final	FROM 1/9/82 TO 31/8/83	AUG 31, 1983	55

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

During this period, the principal investigator studied stability robustness of feedback control systems. Towards this direction, first a new stability robustness condition is developed in time domain (in terms of eigenvalues) and it is shown that the proposed time domain condition is less conservative than the corresponding frequency domain condition as well as another recently developed time domain condition. Also, further observations were made in the comparison of proposed time domain development to the frequency domain development. Then new measures of 'stability robustness' and 'performance robustness' were developed and incorporated into the robust control design algorithm proposed in the summer research.

DTIC FILE COPY

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☒ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Dr. Joseph Bram	(202) 767- 4939	NM

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

84 02 10 136

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

ITEM #11, TITLE: TIME DOMAIN ANALYSIS AND SYNTHESIS OF ROBUST CONTROLLERS FOR LARGE SCALE LQG REGULATORS

AFOSR-TR- 84-0040

Time Domain Analysis and Synthesis of Robust Controllers for Large Scale LQG Regulators

Final Report

of

Mini Grant Research

for

U.S. Air Force Office of Scientific Research

AFOSR-83-0139

By

R.K. Yedavalli*

and

R.N. Shanbhag+ and J. Irudayasamy+

Department of Mechanical Engineering

Stevens Institute of Technology

Hoboken, N.J. 07030

* Assistant Professor

\+ Graduate Students

84 02 10 136

Contents

Time Domain Analysis and Synthesis of Robust Controllers

for Large Scale LQG Regulators

Final Report for AFOSR Minigrant

ABSTRACT

The aspect of 'Robustness' for linear multivariable systems is analyzed in time domain. Both 'Stability Robustness' and 'Performance Robustness' are combinedly considered to meet stability and performance requirements. First a stability robustness condition in time domain (in terms of eigenvalues) is presented and examples are given which indicate that the proposed robustness condition is less conservative then the corresponding frequency domain condition as well as another recently proposed time domain condition, both given in terms of singular values. Next a technique is presented to further reduce the 'conservatism' of the proposed condition. A design algorithm that incorporates both 'stability robustness' and 'performance robustness' into the design procedure suggested in the summer faculty program report, is modified with the help of new definitions of robustness indices. Computer software to implement the algorithm is presented along with simple examples to illustrate the concepts. Based on the experience gained by the minigrant research, areas of future research are recommended.

Nomenclature

R^{α}	=	Real Vector Space of Dimension α
δ	=	Dirac Delta
ε	=	Belongs to
ω	=	Frequency Variable
ρ [.]	=	Spectral radius of the matrix [.]
	=	The largest of the modulus of the eigenvalues of [.]
σ [·]	=	Singular values of the matrix [.]
λ [.]	=	Eigenvalues of the matrix [.]
$[.]_s$	=	Symmetric part of a matrix [.]
$\|[.]\|$	=	Modulus Matrix = Matrix with modulus entries
$\forall$ i	=	for all i

I. INTRODUCTION AND OBJECTIVES

It is well known that the inaccuracies in the mathematical models of physical systems can severely compromise the resulting control designs. The errors associated with mathematical models of physical systems may be broadly categorized as i) parameter errors, ii) truncated models (errors in model order), iii) neglected or incorrectly modeled external disturbances and iv) neglected nonlinearities. It is the inevitable presence of these errors in the model used for design that eventually limits the performance attainable from the control system designs produced by either classical (frequency domain) or modern (time domain) control theory. The problem of model errors is more critical, in general, for large scale Linear Quadratic Gaussian (LQG) optimal control problems and in particular, for aerospace applications like Large Space Structure (LSS) control [1] and other aeroelastic systems [2]. These applications are, of course, of extreme importance to the U.S. Air Force. The fundamental problem of these Distributed Parameter Systems (DPS) control is the control of a large dimensional system with a controller of much smaller dimension (model/controller truncation) compounded with modal data uncertainty (parameter errors). In the light of these observations, it is evident that 'robustness' is an extremely desirable (sometimes, necessary) feature of any feedback control design proposed for DPS control. 'Robustness' studies of Large Scale LQG regulators is the central theme, of the present research.

For our present purposes a 'robust' control design is that design which behaves in an 'acceptable' fashion (i.e. satisfactorily meets the system specifications) even in the presence of modeling errors. Since the

system specifications could be in terms of stability and/or performance (regulation, time response, etc.) we can conceive two types of robustness, namely, 'Stability Robustness' and 'Performance Robustness'. Limiting our attention in this research to 'parameter errors' and 'model/controller truncation' as the two types of modeling errors that may cause instability (or performance degradation) in the system, we fornally define 'stability robustness' and performance robustness' as follows:

'Stability Robustness': Maintaining closed loop system stability in the presence of modeling errors mainly parameter variations and model/controller truncation.

'Performance Robustness': Maintaining satisfactory level of performance in the presence of modeling errors mainly parameter variations and model/controller truncation.

Implicit in the definition of 'Performance Robustness' is the requirement of 'stability' for performance robustness studies. However, performance robustness studies which require (or assume) stability may only have limited application. On the other hand concentrating design efforts on 'stability' alone is not prudent because the performance requirements may not be met by that design. Thus, simultaneous consideration of stability and performance in the design process is more appropriate. Most of the current published literature addresses either the ' stability robustness' aspect or the 'performance robustness' robustness' aspect separately. Most of the interesting work on 'stability robustness' is done in frequency domain using singular value decomposition [3-11] while much of the useful research on 'performance robustness' is carried out in time domain using sensitivity approaches [12-14].

Design studies that treated the stability robustness aspect in time domain and studies which combined both stability robustness and performance robustness into the design process have been scarce. Towards this direction, research was initiated on these aspects by the author during the Summer Faculty Research Program period (Summer '82) and consequently, a stability robustness condition in time domain and a design algorithim that incorporates both stability robustness and performace robustness into the design process were proposed [15]. Later as part of Mini grant work the following research was proposed.

i) To probe further into the possible refinement of the stability condition.

ii) To develop computer software for automating the design algorithim and finally

iii) To illustrate the methodology by examples representative of Large Space Structure models.

In what follows, a summary of the research that accomplished the above tasks is presented followed by a discussion of areas of further research.

II. Work Done During the Mini Grant Period

IIA. Time Domain Analysis of Stability Robustness

Much of the published literature treats stability robustness of feedback control systems in the frequency domain with the help of singular value decomposition [3-11 & 16]. While the singular value analysis is a useful tool to generalize the Nyquist criterion for the multi-variable case, it is not the only way to characterize the stability of a perturbed system,

particularly for systems described by state space models. For systems described by state space differential equations, the analysis of stability in the time domain becomes a viable alternative. The time domain robustness analysis may have a number of advantages over frequency domain treatment as explained in later sections. The time domain treatment is more or less analogous to the frequency domain treatment in spirit, but with the stability conditions given in terms of eigenvalues rather than singular values since eigenvalue analysis is more appropriate for time domain stability assessment. As stated earlier, it is of interest to note that eigenvalue analysis was used even in the frequency domain treatment (Ref. [16]).

In what follows, we first present the main mathematical result (as a new theorem) which forms the basis for developing a condition for the stability of a perturbed matrix. This result is then extended to the case of a linear system. These results are further specialized to the case of LQG regulators with perturbations in the form of i) parameter variations and ii) truncated models.

Main Result:

Let F and E be two real square matrices.

Theorem 1: If F is negative definite, then the matrix F + E is negative definite if

$$\rho\{[E_s(F_s)^{-1}]_s\} < 1 \tag{1}$$

Proof: Given in Appendix A.

The above theorem has an interesting implication. It suggests that if

a given matrix F_1 is written as the sum of a negative definite matrix F and a perturbation matrix E

i.e., $F_1 = F + E$ where F is negative definite

then by use of theorem 1 one can get a condition for the negative definiteness and hence the stability of matrix F_1.

Application to Linear State Space Models:

Let us consider the linear time invariant system

$$\dot{x} = A x \quad , \quad x(0) = x_0 \quad , \quad x \in R^n \tag{2}$$

where the 'nominal' matrix A is asymptotically stable. Let there be an additive perturbation E in the matrix A so that the perturbed system matrix is A + E. We are now interested in the stability of the matrix A+E. The straightforward application of theorem 1 gives the following result.

Theorem A: The perturbed system matrix A+E is stable if

$$\rho\,[(E_n A_d^{-1})_s] < 1 \tag{3}$$

where $A = A_d + A_e$., $A_d = \text{Diag}\,[\alpha_i]$, $i=1,2...n$, α_i is any real negative entry ($\alpha_i < 0$) and $E_n = (A_e + E)_s$.

Proof: Since A is asymptotically stable, we can always write

$$A = A_d + A_e \tag{4}$$

where $A_d = \text{Diag}\,[\alpha_i]$, $i = 1,2,...n.$, α_i is real and < 0.

thus $A + E = A_d + (A_e + E)$

where A_d is a symmetric negative definite matrix. Applying theorem 1 gives the result of theorem A since a negative definite matrix is always a stability matrix.

At this stage, it is appropriate to comment on the applicability of these stability conditions. It is to be noted that the derived stability condition is not particularly useful when one knows both the matrices A and the perturbation matrix E, in which case the stability of the matrix $A + E = A_d + A_e + E$ is determined by simply looking at its eigenvalues. However, in a practical situation, one doesn't exactly know E. One may only have knowledge of the magnitude of the maximum deviation that can be expected in the entries of A. In that case the entries of E are such that

$$|E_{ij}| \leq \Delta_{ij} \tag{C1}$$

where Δ_{ij} is the magnitude of the maximum deviation.

The following theorem enhances the usefulness of the proposed stability criterion.

<u>Theorem B</u>: The perturbed system matrix A+E is stable for <u>all</u> perturbations E_{ij} satisfying (C1) if

$$\rho\,[(E_m A_m^{-1})_s] < 1 \tag{5}$$

where $A = A_d + A_e$, $A_d = \text{Diag}\,[\alpha_i]$, $i = 1,2,\ldots n, \alpha_i < 0$

$$E_m = (|A_e| + \Delta)_s \text{ and } A_m = |A_d| \tag{6}$$

<u>Proof</u>: Observe that

$$\rho[\{(|A_e| + \Delta)_s\, |A_d|^{-1}\}_s] > \rho[|\{(A_e + E)_s (A_d)^{-1}\}_s|] > \rho[\{(A_e + E)_s (A_d)^{-1}\}_s]$$

(for all E_{ij} satisfying (C1)). (7)

This follows from the fact that

i) for any given square matrix F

$$\rho\ (|F|) \geq \rho\ (F) \qquad \text{(Ref. [16])} \tag{8}$$

ii) for two given square non-negative matrices F_1 and F_2 such that $F_{1ij} \geq F_{2ij}$ for all i,j

$$\rho\ (F_1) \geq \rho\ (F_2) \qquad \text{(Ref. [17])} \tag{9}$$

thus

$$\rho\ [(E_m A_m^{-1})_s] < 1 \rightarrow \rho\ [(E_n A_d^{-1})_s] < 1 \tag{10}$$

which in turn implies A+E is stable for <u>all</u> E_{ij} satisfying (C1).

Q.E.D.

Thus, in practice, we test the condition of theorem B and if satisfied we can guarantee asymptotic stability for all perturbations E_{ij} satisfying (C1).

Note that there is some flexibility in splitting the stable matrix A into A_d and A_e. One immediate choice for A_d could be

$$A_d = \text{Re}\ [\lambda_i(A)],\ i = 1,2,\ldots.n \tag{C2}$$

We now compare the ability of the proposed time domain condition in predicting the stability of a perturbed system with i) the corresponding frequency domain condition and ii) another time domain condition recently proposed, both given in terms of singular values.

Recently, in Ref. [5], Lee et al. considered the state feedback regulator of Fig. 1 and provided a condition for the stability of an additively perturbed system. It is shown that for a system which is

initially stable, stability will be maintained as perturbations ΔA are added, provided that

$$\sigma_{min}[j\omega I - A_{CL}] > \sigma_{max}(\Delta A) \text{ for all } \omega > o \tag{11}$$

where $A_{CL} = A - BK$ is an asymptotically stable matrix.

The proposed time domain condition for this case takes the form

$$\rho\ [(E_m A_m^{-1})_s] < 1 \tag{12}$$

where $A_{CL} = A_d + A_e$, $A_d = \text{Diag}\ [\text{Re}\ \lambda_i\ (A_{CL})]$, $i = 1,2,\ldots n$

$E_m = (|A_e| + |\Delta A|)_s$ and $A_m = |A_d|$ (i.e. we let A_d take the form (13) given in (C2)).

The following example shows that the proposed time domain condition is less conservative than the frequency domain condition of (11).

<u>Example 1</u>: Let $A_{CL} = \begin{bmatrix} -8 & 0 \\ 0 & -1 \end{bmatrix}$, $\Delta A = \begin{bmatrix} 0 & 1.5 \\ 1.5 & 0 \end{bmatrix}$

<u>Freq. Domain Condition</u>:

$$\sigma_{min}[j\omega I - A_{CL}] = \sigma_{min} \begin{bmatrix} j\omega + 8 & 0 \\ 0 & j\omega + 1 \end{bmatrix} = \sqrt{1 + \omega^2}$$

$$\sigma_{max}\ [\Delta A] = 1.5$$

at $\omega = 1$

$$\sigma_{min}\ [j\omega I - A_{CL}] = 1.414 \not> \sigma_{max}\ [\Delta A] \tag{14}$$

Thus the frequency domain condition fails to predict the stability of the

perturbed system matrix $A_{CL} + \Delta A$.

Proposed Time Domain Condition:

$$A_d = A_{CL} \text{ and thus } E_m = \begin{bmatrix} 0 & 1.5 \\ 1.5 & 0 \end{bmatrix} \quad \text{and } A_m = \begin{bmatrix} 8 & 0 \\ 0 & 1 \end{bmatrix}$$

so

$$\rho\,[(E_m A_m^{-1})_s] = \rho \begin{bmatrix} 0 & 0.84375 \\ 0.84375 & 0 \end{bmatrix} = 0.84375 < 1 \tag{15}$$

Thus the perturbed system matrix $A_{CL} + \Delta A$ is guaranteed to be stable and it is indeed stable.

Comparison with the time domain condition of Lee [18]:

In [18], Lee proposes the following stability robustness condition in time domain in terms of the singular values. The perturbed system matrix $A + \Delta A$ is stable if

$$\sigma_{max}(\Delta A) < -\sigma_{min}(A) \cos(\Theta_{min})$$

where Θ_{min} is the smallest principal phase of A measured counterclockwise from the positive real axis. The above theorem is stated under the assumption that the orthogonal matrix U in the polar decomposition of A, viz

$$A = U H_R \text{ or } A = H_L U$$

is a stable matrix. Specializing this condition for a symmetric stable matrix, we get the condition as

$$\sigma_{max}(\Delta A) < |\lambda_i(A)|_{min}$$

Example 2: Let us consider the same example as before. The time domain condition of (12) yields

$$\sigma_{max}[\Delta A] = 1.5 \nless |\lambda_i(A)|_{min} = 1$$

Thus the above condition fails to predict the stability whereas (as shown in Example 1) the proposed time domain condition succeeds.

Also note that the proposed time domain condition of this research doesn't require the assumption of the orthogonal matrix U being stable which involves testing of yet another condition [18].

Improvement of 'Optimism' of the proposed condition:

The flexibility in splitting the stable matrix A into A_d and A_e can be utilized to further improve the optimism of the proposed condition.

One suggested technique is as follows:

Procedure: 1) Write $A_d = -\alpha I$

where $\alpha>o$ is a scalar variable.

The stability criterion then becomes

$$\rho\left(\frac{E_m}{\alpha}\right) < 1$$

where $E_m = (|A_e| + |E|)_s$.

The above condition is then tested for various values of α which clearly increases the probability of guaranteeing the stability of a perturbed system whenever it is stable.

Example 3: Let $A_{CL} = \begin{bmatrix} -8 & 0 \\ 0 & -1 \end{bmatrix}$; $\Delta A = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$

with A_d = Diag $[\lambda_i(A_{CL})]$, the proposed time domain condition becomes

$$\rho \begin{bmatrix} 0 & 2 \\ 0.25 & 0 \end{bmatrix}_s = \rho \begin{bmatrix} 0 & 1.125 \\ 1.125 & 0 \end{bmatrix} = 1.125 \nless 1$$

But with $A_d = -\alpha I$, the test matrix for the proposed time domain condition becomes

$$\rho \begin{bmatrix} \frac{|\alpha-8|}{\alpha} & 2/\alpha \\ 2/\alpha & \frac{|(\alpha-1)|}{\alpha} \end{bmatrix} = ?$$

choosing $\alpha = 8$, we get

$$\rho \begin{bmatrix} 0 & 0.25 \\ 0.25 & 0.875 \end{bmatrix} = 0.9414 < 1 \rightarrow \text{the system is stable.}$$

Thus, the proposed time domain condition with 'α splitting' is able to predict the stability of the perturbed system and thus yields a less conservative result.

Evidently, there is much scope to improve the 'optimism' of the proposed time domain condition by an appropriate selection of the A_d matrix and more research is warranted in this direction.

Extension to LQG Regulators

We now extend the above analysis to the case of large scale LQG regulators having i) parameter variations and ii) truncated modes as the modeling errors (or perturbations). The strategy adopted is to model these perturbations as additive perturbations to a nominally stable matrix and then

apply the above eigenvalue analysis to arrive at the stability conditions.

Let us consider a continuous linear time invariant system described by

$$\dot{x}\ (t) = A\ x\ (t) + B\ u\ (t) + D\ w\ (t) \quad , x(0) = x_0 \tag{16a}$$

$$y\ (t) = C\ x\ (t) \tag{16b}$$

$$z\ (t) = M\ x\ (t) + v\ (t) \tag{16c}$$

where the state vector x is nx1, the control u is mx1, the external disturbance w is qx1, the output y (the variables we wish to control) is kx1 and the measurement vector z is ℓx1. Accordingly, the matrix A is of dimension nxn, B is nxm, D is nxq, C is kxn and M is ℓxn. The initial condition x(0) is assumed to be a zero-mean, gaussian random vector with variance X_o, i.e.

$$E[x(0)] = 0, \qquad E[x(0)\ x^T(0)] = X_o \tag{17}$$

Similarly, the process noise w (t) and the measurement noise v (t) are assumed to be zero-mean white noise processes with gaussian distributions having constant covariances W and V respectively, i.e.

$$E[w(t)] = E[v(t)] = 0 \tag{18}$$

$$E \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} [w^T(\tau) \quad v^T(\tau)] =$$

$$\begin{bmatrix} W & 0 \\ 0 & \rho_e V_o \end{bmatrix} \delta\ (t - \tau) \tag{19}$$

where ρ_e is a scalar greater than zero . . . and $V = \rho_e V_o$.

Let the above system be evaluated for any control u by the quadratic performance index

$$J = \lim_{t \to \infty} \frac{1}{t} \; E \int_o^t [(y^T(\tau)\ Q\ y\ (\tau) + u^T(\tau) \rho_c R_o\ u\ (\tau)]\ d\tau \tag{20}$$

where scalar $\rho_c > 0$ and Q, R_o are (kxk) and (mxm) symmetric, positive definitive matrices, respectively.

For the case of a deterministic system, the following modifications in the system description are in order:

i) $Dw = 0, \quad v = 0$

ii) the initial condition, $x(0) = x_o, x_o x_o^T = X_o$

and the index J of (20) reads

$$J = \int_o^\infty [y^T(t)\ Q\ y(t) + u^T\ (t)\ \rho_c R_o\ u(t)] dt \tag{21}$$

If the state x(t) of the stochastic system is estimated as a function of the measurements we assume the state estimator to be of the following structure

$$\dot{\hat{x}}(t) = A\,\hat{x}(t) + \hat{G}\,\hat{z}(t) \tag{22}$$

where

$$\hat{z}(t) = z(t) - M\,\hat{x}(t)$$

is called the 'measurement residual'. For a 'minimum variance' requirement, the estimator of (22) is the standard Kalman filter [19]. We refer to the system presented in this section as the 'Basic System'.

We now use theorem B as the basis for extending the stability conditions to the case of LQG regulators for various cases of modeling errors involving parameter variations and model/controller truncation. These are discussed in [15]. For brevity we consider two cases here.

Case 1: Parameter variations alone, no model/controller truncation:

For this case, the following assumptions are made with respect to the model described by equations (16).

Assumption 1: The matrix pairs [A , B] and [A , D] are completely controllable and the pairs [A , C] and [A , M] are completely observable.

Since there is no model/controller truncation the full order optimal control for nominal values of the parameters is given by

$$u = G\,\hat{x} = -\frac{1}{\rho_c} R_o^{-1} B^T K\,\hat{x} \tag{23a}$$

where

$$\dot{\hat{x}} = A\hat{x} + Bu + \hat{G}(z - M\hat{x}), \quad \hat{x}(0) = 0 \tag{23b}$$

$$= (A + BG - \hat{G}M)\hat{x} + \hat{G}z \tag{23c}$$

$$\hat{G} = \frac{1}{\rho_e} P M^T V_o^{-1} \tag{23d}$$

and P and K satisfy the algebraic matrix Riccati equations

$$KA + A^T K - KB \frac{R_o^{-1}}{\rho_c} B^T K + C^T QC = 0 \tag{23e}$$

$$PA^T + AP - PM^T \frac{V_o^{-1}}{\rho_e} MP + DWD^T = 0 \tag{23f}$$

The nominal closed loop system is given by

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & BG \\ \hat{G}M & \hat{A}_c \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} D & 0 \\ 0 & \hat{G} \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} \tag{24a}$$

$$\begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \tag{24b}$$

where $\hat{A}_c = A + BG - \hat{G}M$ and the closed-loop system is asymptotically stable.

We are now interested in examining the stability robustness of the closed-loop system in the presence of parameter variations alone. Let ΔA, ΔB, ΔC, ΔM and ΔD be the maximum modulus perturbations in the system matrices, A, B, C, M and D respectively. Then the perturbed

system can be written as (Note: the filter parameters $\hat{A}_c$ and $\hat{G}$ and the control gain G are not subjected to variations because they are specified by the designer as a function of the nominal values of the parameters).

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \\ \hline \Delta\dot{x} \\ \Delta\dot{\hat{x}} \end{bmatrix} = \left[\begin{array}{cc|cc} A & BG & 0 & 0 \\ \hat{G}M & \hat{A}_c & 0 & 0 \\ \hline \Delta A & \Delta BG & A+\Delta A & (B+\Delta B)G \\ \hat{G}\Delta M & 0 & \hat{G}(M+\Delta M) & \hat{A}_c \end{array}\right] \begin{bmatrix} x \\ \hat{x} \\ \hline \Delta x \\ \Delta\hat{x} \end{bmatrix} + \left[\begin{array}{cc} D & 0 \\ 0 & \hat{G} \\ \hline \Delta D & 0 \\ 0 & 0 \end{array}\right] \begin{bmatrix} w \\ v \end{bmatrix} \tag{25}$$

By application of theorem B of section II, we obtain the following design observation.

Design Observation 1: The perturbed LQG regulator system is stable for all perturbations in A,B,C,M & D in the sense of (C1) if

$$\rho[(E_m A_m^{-1})_s] < 1 \tag{26}$$

where

$$A_d = \text{Diag }[\text{Real } \lambda_i(A_{CL})] \text{ , } A_{CL} = \begin{bmatrix} A & BG \\ \hat{G}M & \hat{A}_c \end{bmatrix}$$

$$A_{CL} = A_d + A_e \text{ , } E_m = |A_e| + E, \ E = \begin{bmatrix} \Delta A & \Delta B|G| \\ |\hat{G}|\Delta M & 0 \end{bmatrix} \text{and } A_m = |A_d| \tag{27}$$

Note that in this case both the nominally stable matrix A_{CL} and the perturbation matrix E are functions of controller gains G and $\hat{G}$.

Case 2: Model reduction alone, no controller reduction and no parameter variations

For this case, we treat the model given by (16) as the evaluation model. We assume the order of the model, n, to be too high for the control u to be determined and that there is a control design model of dimension $n_R < n$ given by

$$\dot{\bar{x}}_R = \bar{A}_R\bar{x}_R + \bar{B}_R u + \bar{D}_R w, \quad \bar{x}_R \varepsilon R^{n_R}$$

$$\bar{y}_R = \bar{C}_R\bar{x}_R \tag{28}$$

$$\bar{z}_R = \bar{M}_R\bar{x}_R + v$$

where the above control design model is obtained either by a direct truncation of the full order model given by

$$\dot{x} = \begin{bmatrix} \dot{x}_R \\ \dot{x}_T \end{bmatrix} = \begin{bmatrix} A_R & A_{RT} \\ A_{TR} & A_T \end{bmatrix} \begin{bmatrix} x_R \\ x_T \end{bmatrix} + \begin{bmatrix} B_R \\ B_T \end{bmatrix} u + \begin{bmatrix} D_R \\ D_T \end{bmatrix} w \tag{29a}$$

$$y = [C_R \quad C_T] \begin{bmatrix} x_R \\ x_T \end{bmatrix} \tag{29b}$$

$$z = [M_R \quad M_T] \begin{bmatrix} x_R \\ x_T \end{bmatrix} \quad ; \quad x_R \varepsilon R^{n_R} \tag{29c}$$

or by a partial realization of x involving some model reduction technique (e.g. [14]).

Let the full order control for the reduced order model be given by

$$u = \bar{G}_R \bar{\hat{x}}_R \tag{30a}$$

$$\dot{\bar{\hat{x}}}_R = \bar{A}_R\bar{\hat{x}}_R + \bar{B}_R u + \bar{\hat{G}}_R(z - \bar{M}_R\bar{\hat{x}}_R) \tag{30b}$$

$$= \bar{\hat{A}}_R \bar{\hat{x}}_R + \bar{\hat{G}}_R z \text{ where } \bar{\hat{A}}_R = \bar{A}_R + \bar{B}_R\bar{G}_R - \bar{\hat{G}}_R\bar{M}_R$$

such that the closed-loop system matrix for the control design model given by

$$A_{cL} = \begin{bmatrix} \bar{A}_R & \bar{B}_R\bar{G}_R \\ \bar{\hat{G}}_R\bar{M}_R & \bar{\hat{A}}_R \end{bmatrix} \tag{31}$$

is asymptotically stable. These controller gains $\bar{G}_R$ and $\bar{\hat{G}}_R$ could be optimal or non-optimal with respect to the model (28).

The closed-loop system for the evaluation model is obtained by forcing the evaluation model with the controller of the control design model. Thus, we have

$$\begin{bmatrix} \dot{x}_R \\ \dot{\bar{\hat{x}}}_R \\ \dot{x}_T \end{bmatrix} = \begin{bmatrix} A_R & B_R\bar{G}_R & A_{RT} \\ \bar{\hat{G}}_R M_R & \bar{\hat{A}}_R & \bar{\hat{G}}_R M_T \\ A_{TR} & B_T\bar{G}_R & A_T \end{bmatrix} \begin{bmatrix} x_R \\ \bar{\hat{x}}_R \\ x_T \end{bmatrix} + \begin{bmatrix} D_R & 0 \\ 0 & \bar{\hat{G}}_R \\ D_T & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} \tag{32}$$

$$\begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} C_R & 0 & C_T \\ 0 & \overline{G}_R & 0 \end{bmatrix} \begin{bmatrix} x_R \\ \bar{\hat{x}}_R \\ x_T \end{bmatrix}$$

The stability of the above closed-loop system matrix which we denote as A_{CL2} is to be established.

At this juncture we assume the matrix A_T to be an asymptotically stable matrix (which is a reasonable assumption for large space structure models). In order to derive the condition for stability of the closed-loop system matrix A_{CL2}, we write A_{CL2} of (32) as

$$A_{CL2} = \begin{bmatrix} A_R & B_R G_R & 0 \\ \hat{G}_R M_R & \hat{A}_R & 0 \\ 0 & 0 & A_T \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & B_R(\overline{G}_R - G_R) & A_{RT} \\ (\bar{\hat{G}}_R - \hat{G}_R) M_R & \bar{\hat{A}}_R - \hat{A}_R & \bar{\hat{G}}_R M_T \\ A_{TR} & B_T G_R & 0 \end{bmatrix} \tag{33}$$

$$= A_{s.c} + E_{F.L.} \qquad , \quad \hat{A}_R = A_R + B_R G_R - \hat{G}_R M_R$$

where G_R and $\hat{G}_R$ are the control gain and estimator gain matrices, respectively, obtained by using the reduced order model $\{A_{R,} B_{R,} C_{R,} M_{R,} D_{R,}\}$, and are such that the resulting closed loop system matrix (of the reduced order system which, of course, is the first partition of $A_{s.c}$) is asymptotically stable. One choice of G_R and $\hat{G}_R$ could be the standard optimal control gains of the reduced order model $\{A_{R,} B_{R,} C_{R,} M_{R,} D_{R,}\}$ under appropriate conditions. Note that $A_{s.c}$ is thus a stable matrix and that $E_{F.L}$ basically contains the terms that cause spillover.

<u>Design Observation 2</u>: The closed-loop system matrix A_{CL2} is stable for <u>all</u> control design models $\overline{G}_R$ and $\overline{\hat{G}}_R$ such that

$$|\overline{G}_{Rij}| \leq |G_{Rij}|, \; |\overline{\hat{G}}_{Rij}| < |\hat{G}_{Rij}|$$

if

$$\rho\,[(E_m A_m^{-1})_s] < 1 \tag{34}$$

where $A_d + A_e = A_{s.c}$, $A_d = \mathrm{Diag}\,[\mathrm{Re}\,\lambda_i\,(A_{s.c})]$

$$E_m = (|A_e| + \Delta E)_s \,, \quad A_m = |A_d|$$

and

$$\Delta E = \begin{bmatrix} 0 & |2B_R G_R| & |A_{RT}| \\ |2\hat{G}_R M_R| & |2\hat{A}_R| & |\hat{G}_R M_T| \\ |A_{TR}| & |B_T G_R| & 0 \end{bmatrix} \tag{35}$$

Measure of Stability Robustness

It may be noted that a variety of controller gains may satisfy the proposed stability condition for given perturbations. In order to compare different models/controllers from stability robustness point of view, it is clear that there is a need for some form of a measure of stability robustness. To this end, we now define a measure of stability robustness called "Stability Robustness Index", $\beta_{S.R.}$

$$\beta_{S.R.} \triangleq \left|\, |\mathrm{Re}\,\lambda_{min}(A_{CL})| - |\mathrm{Re}\,\lambda_{min}(A_{CLP})| \,\right| \Big/ |\mathrm{Re}\lambda_{min}(A_{CL})| \tag{36}$$

where A_{CL} is the nominal closed loop system matrix and A_{CLP} is the perturbed closed loop system matrix (i.e. the closed loop system matrix formed by the perturbed matrices of plant, estimator, etc.); here it is assumed that the controller gains are such that the condition for stability is satisfied and thus the perturbed closed loop system matrix is stable. The motivation behind this definition is that the index is a measure of the magnitude of the deviation in the minimum eigenvalue modulus of the perturbed system from the nominal system.

By this difinition $\beta_{S.R.} = 0$ corresponds to a highly robust system from stability point of view. However, one aspect of further research in this development is to investigate what is the worst case deviation (i.e. Max $\beta_{S.R.}$) for a given set of maximum modulus perturbations expected in the system matrices. Once the worst case $\beta_{S.R.}$ is determined for each controller/model, say $\beta_{S.R.W.}$ then this index can be used for comparison purposes.

Measure of Performance Robustness

In similar lines, we define a measure of performance robustness called "Performance Robustness Index $\beta_{P.R.}$" i.e.

$$\beta_{P.R.} \triangleq \left| J_P - J_N \right| \Big/ J_N \tag{37}$$

where J_P is the value of the performance index for the perturbed systen (with A_{CLP} as the plant matrix). Thus, $\beta_{P.R.} = 0$ corresponds to a highly robust system from performance point of view. As mentioned above, an area of future research is to investigate the worst case $\beta_{P.R.}$ for each controller/ model and then use this index $\beta_{P.R.W.}$ as the basis for comparison.

Robust Control Design

Once measures of stability robustness and performance robustness are developed, the idea of a robust control design is to pick a controller that gives a reasonable or satisfactory trade off between stability and performance. Specifically, the design algorithms involves determining the indices $\beta_{S.R.}$ and $\beta_{P.R.}$ for different values of the design parameters ρ_c and ρ_e which in turn determine the control and estimator gains. This information about the indices $\beta_{S.R.}$ and $\beta_{P.R.}$ along with the corresponding nominal costs $({}_y T_y)^{\frac{1}{2}}$ and $({}_u T_u)^{\frac{1}{2}}$ can be used to pick a specific controller (control and estimator gain combination) as the one which provides a satisfactory trade off between stability and performance. The algorithm is thus iterative in nature. The computation basically involves the use of matrix Riccati and Liapunor solutions and the eigenvalue analysis for which standard easy-to-use computer programs are available. The details of the algorithms (in principle) are given the SFRP report and for reasons of brevity, are not reported here. However, a brief account of the "flow-chart" is included in the section dealing with the computer software.

Discussion of the Theoretical Development of this Research:

Some discussion about the implications of the current theoretical development of this research is now in order. First, it may be noted that the proposed stability conditions are similar, conceptually, to the frequency domain results reported in Ref. [1]. However, there are also some interesting differences between these two (frequency domain and time domain) versions. Some preliminary observations are presented in the following sections. Secondly, these design observations are useful in many ways in both the analysis and synthesis of robust controllers. These are discussed in later sections.

Comparison and Contrast Between Frequency Domain Analysis and the Time Domain Analysis

The main differences between the frequency domain treatment and the time domain treatment are as follows:

i) The main differences between the frequency domain treatment and the the calculation of singular values of a complex matrix at various frequencies. In the stability conditions of time domain, no time dependence is present. Only the eigenvalues of a real symmetric matrix are to be computed.

ii) In the case of frequency domain results, the perturbations are mainly viewed in terms of 'gain' and 'phase' changes [6,7]. In the proposed time domain analysis the perturbations are viewed as 'system parameter variations' and 'system model/controller order' with constant, fixed gains. It may be noted that in the time domain treatment the nominally stable closed-loop matrix and the perturbed closed-loop matrix are both functions of the constant controller gains.

iii) In the frequency domain treatment, the requirement of square matrices necessitates the assumption that the number of control inputs be equal to the number of outputs, which can be satisfied by appropriate selection of the break point of the loop. However, in the present time domain analysis no such assumption is needed.

iv) Most of the work on robustness in the frequency domain concentrated on the analysis problem. That is, analyzing a given closed-loop control system to determine the uncertainty bounds that will make the closed-loop system unstable. This can be done very easily in the frequency domain. The difficulty, however, is that the uncertainty bounds can be obtained only in the frequency domain and are very difficult to translate back into the time domain to determine the allowable perturbations of physical parameters of the system. This difficulty can be eliminated by posing equations (26) and (27) in the framework of a robustness analysis problem.

Very little work has been done on the robustness synthesis problem in the frequency domain. That is, to design a closed-loop contorl system to meet the performance specifications in the face of a specified set of uncertainty bounds. The proposed approach in the time domain handles the robustness synthesis problem very well.

v) In the proposed stability conditions, provision is made for considering different reduced order models for control design purposes (reflected by the presence of $\overline{A}_R$, $\overline{B}_R$ matrices which could be different from A_R,B_R...matrices). Just as in the frequency domain different control design methods could be compared using the singular value plots of the return-difference matrices, this provision helps to compare different reduced order models (used for control design) from a stability robustness point of view.

vi) In the frequency domain treatment, considering an uncertainty, for example, as an additive perturbation several stability robustness conditions can be written which do not imply each other for practical systems [21]. In the present time domain approach such difficulty is not present. The perturbations can only be modelled as additive perturbations and yield only one robustness test.

vii) All the norm bounded robustness criteria in the frequency domain are inverently conservative because they assume that the phases of all the elements in the perturbation matrix are in worst possible direction which is a mathematical extreme. Some possible alternatives to reduce this conservatism in the frequency domain are developed in references [22] and [23]. The proposed time domain robustness criteria are also conservative. The conservatism enters the development of the main result because of the requirement of the negative definitess of the perturbed system. The degree of conservatism, however, is a subject for future research.

These are some of the preliminary observations made with respect to the frequency domain and time domain approaches for 'stability robustness'. Evidently further in-roads have to be made in the investigation of this relationship and this is suggested as a future-research topic. In the following section, the usefulness of the proposed design observations is briefly discussed.

Usefulness of the Design Observations

The proposed design observations are helpful in many ways. First, if a nominally stable closed-loop system matrix is specified (i.e. either A_{CL} or $A_{s.c}$ depending on the specific case as discussed in section III), then one can use these tests to determine the tolerable perturbations in the

system matrices A, B and M and the order of the model/controller before the closed loop system becomes unstable.

Conversely, given the perturbations ΔA, ΔB and ΔM and the order of the model/controller, one may determine the controller gains to achieve stability robustness.

As indicated in the previous section, these tests can be used to compare different model reduction and control design schemes from a stability robustness point of view.

Finally, these tests can find applications in spillover reduction problems and sensor/actuator location problems.

IIB. Computer Software

Introduction:

The computer program has been developed in two packages. The first package tests the stability condition developed in the report for two types of perturbations viz, Parameter Variation and Model truncation. The second package then computes the Regulation, control and total costs for the open loop. Nominal closed loop and Perturbed Closed Loop Systems. The program then computes the robustness indices $\beta_{P.R.}$ and $\beta_{S.R.}$.

The program uses extensively the subroutines available in the LSLIB (Library for Control and Estimation of Linear Uncertain Systems) and made compatible for use in DEC-10 computer. The main programs of the Stevens Computer Center and other subroutines are written in Fartran IV. The LSLIB package was originally developed at Purdue Aero Department.

The following paragraphs outline briefly the program.

Package 1. Stability Evaluation:

The algorithm first forms the nominal closed loop matrix A_{CL} and the Perturbation Matrix ΔA_{CL}.

a) Program PARVA 1 handles parameter variation problem. The input matrices are A,B,C,D,M,Q,R,V,W and the perturbations DA,DB,DC,DD&DM.

1) The subroutine SSLQG solves the algebraic matrix Rocatti equation for the controller and estimator and gives the control gain G and estimator gain $\hat{G}$.

2) The subroutine MATFTZ forms the nominal closed loop matrix A_{CL} and the Perturbation Matrix ΔA_{CL}.

b) Program Trunk handles the Truncated Mode problem. The input matrices are A_R, A_{RT}, A_{TR}, B_R, B_T, C_R, C_T, D_R, D_T, M_R, M_T, Q, R, V, W.

1) The subroutine SSLQG solves the Ricatti equation and gives the control gain G_R and estimator gain $\hat{G}_R$.

2) The subroutine MATFIT forms the nominal closed loop Matrix A_{CL} and the Perturbation matrix ΔA_{CL} as discussed in the report.

c) Program TEST with A_{CL} and ΔA_{CL} matrices as input, tests the stability condition of the system. It is interactive and permits one to test the stability of the system for different values of α (of the α method discussed in the report). Once the condition is satisfied, the program then calculates the stability-robustness index $\beta_{S.R.}$ of (36) for all the controllers satisfying the condition.

Package 2. Performance Evaluation:

The input to this program are the matrices AB, C, D, M, W, Q, DA, DB, DC, DD, DM and the scalar constants ROEC and ROEE.

1) The subroutine LYAP2 computer the open loop cost J_{op}.

2) Subroutine SSLQG solves the Ricatti equations for the controller and estimator and gives the control gain G and the estimator gain $\hat{G}$. It also computes the control, regulation and total costs of the nominal closed loop system (JUN, JXN, JN respectively).

3) Subroutines MATFT2 and MATFT4 form the matrices AF11, AF21, AF22, CF, DF1, DF2, and WF (Ref. [15].

4) Subroutines LYAP 2 and LYAP 5 solve the three reduced-order Lyapunov equations (each of order $(n + n_c)$) and give the matrices PF11, PF12, and PF22.

5) Subroutine MATFT2 forms the matrix PF.

6) Subroutine MATFT4 forms the matrix QF.

7) Subprogram JRAC2 computes the cost with the matrices PF, CF and QF as input.

8) The algorithm forms different structures of the QF matrix each one yielding a particular cost, thus, the control costs JUNN, JUP, Regulation Costs JXNN, JXP and the total costs JNN and JP for the nominal closed loop and Perturbed Closed Loop respectively are computed.

9) The algorithm then computes the performance robustness Index BETAPR from the perturbed regulation cost JXP and the nominal regulation cost JXNN.

List of Subroutines Used:

ABS	-	Makes all elements of a matrix absolute.
CINV	-	Forms inverse of a complex (nonsingular) matrix.
DIAMAT	-	Forms diagonal matrix with real part of eigenvalues of a given matrix.
EIGRF	-	Computes eigenvalues.
IDENT	-	Forms an identity matrix.
LYAP 2	-	Solves Lyapunov equation [form: $AA * xx + xx * AA^T + cc = 0$]
LYAP 5	-	Solves Lyapunov Equation of the [form: $AA * xx + xx * BB^T + cc = 0$].
MADDSB	-	Adds and subtracts Matrices [form: A + B - c].
MARB	-	Find the biggest magnitude of real part of the eigenvalue.
MARS	-	Find the smallest magnitude of real part of the eigenvalue.

MATFT 2	-	Forms a single matrix from 4 matrices.
MATFT 4	-	Forms a single matrix from 16 matrices.
MEQ	-	Stores the matrix A in the matrix B.
MP 31	-	Computes matrix product [form: $p = A^T * B * C$
MP 32	-	Computes matrix product [form: $p = A * B * C^T$
MULRRT	-	Computes matrix product [form: $p = A * B^T$
MULT	-	Computes matrix product [form: $p = A * B$]
SCAMUL	-	Multiple a matrix by a scalar.
SSLQG	-	Solves Ricattis equation.
STABR	-	Forms and solves the stability criterion.
TRAC 2	-	Multiple 2 matrices and finds the trace of the matrix product.
USWCM	-	Prints complex matrix.
USWFM	-	Prints real matrix.

```
C     PROGRAM PARVA1
C THIS PROGRAM FORMS CLOSED LOOP MATRIX AND PETURBATION MATRIX
C FOR PARAMATER VARIATION PROBLEM
      REAL A(14,14),B(14,14),C(14,14),D(14,14),M(14,14),DA(14,14)
      REAL DB(14,14),DC(14,14),AD(28,28),WW(14,14)
      REAL DD(14,14),DM(14,14),Q(14,14),R(14,14)
      REAL W(14,14),V(14,14),S(14,14),CQC(14,14)
      REAL K(14,14),P(14,14),F(14,14),G(14,14),WK(500),ZERO(14,14)
      REAL FM(14,14),BG(14,14),AC(14,14),AFM(14,14),ABG(14,14)
      REAL ACL(28,28),AE(28,28),DBG(14,14)
      REAL FDM(14,14),ECL(28,28),EN(28,28)
      REAL CC(28,28),DDD(28,28),ENS(28,28)
C
      COMPLEX EIGA(14),EIG(14),EE(28,28),EI(28,28)
      COMPLEX EIGACL(28),EIGN(28)
C
        IS=14
        IB=28
        NX=2
        NM=1
        NQ=1
        NK=2
        NL=2
C
C     GIVE VALUES OF A AND DA
2       READ (15,*)((A(I,J),J=1,NX),I=1,NX)
3       READ (15,*)((DA(I,J),J=1,NX),I=1,NX)
C
C     GIVE VALUES OF B&DB
5       READ (15,*)((B(I,J),J=1,NM),I=1,NX)
6       READ (15,*)((DB(I,J),J=1,NM),I=1,NX)
C
C     GIVE VALUES OF C&DC
8       READ(15,*)((C(I,J),J=1,NX),I=1,NK)
9       READ(15,*)((DC(I,J),J=1,NX),I=1,NK)
C
C     GIVE VALUES OF D&DD
11        READ (15,*)((D(I,J),J=1,NQ),I=1,NX)
121         READ (15,*)((DD(I,J),J=1,NQ),I=1,NX)
C
C
C     GIVE VALUES OF M&DM
14        READ (15,*)((M(I,J),J=1,NX),I=1,NL)
15        READ (15,*)((DM(I,J),J=1,NX),I=1,NL)
C
C     GIVE VALUES OF Q(NK,NK)
C     Q IS SYM,+VE DEFN MATRIX
16        READ(15,*)((Q(I,J),J=1,NK),I=1,NK)
C
C     GIVE VALUES OF ROEC
          READ(15,*)ROEC
          DO  101 I=1,NM
          DO  101 J=1,NM
          R(I,J)=0
101       R(I,I)=1
C     GIVE VALUES OF ROEE
        READ(15,*)ROEE
          DO 102 I=1,NL
          DO 102 J=1,NL
          V(I,J)=0
102       V(I,I)=1
CD          READ(15,*)((V(I,J),J=1,NL),I=1,NL)
C
C CALLS IMSL ROUTINE FOR SETTING OUTPUT IDENTIFIER
          CALL UGETIO(3,0,35)
          CALL USWFM(2HAA,2,A,IS,NX,NX,4)
          CALL USWFM(2HDA,2,DA,IS,NX,NX,4)
          CALL USWFM(2HBB,2,B,IS,NX,NM,4)
          CALL USWFM(2HDB,2,DB,IS,NX,NM,4)
          CALL USWFM(2HCC,2,C,IS,NK,NX,4)
          CALL USWFM(2HDC,2,DC,IS,NK,NX,4)
          CALL USWFM(1HD,1,D,IS,NX,NQ,4)
          CALL USWFM(2HDD,2,DD,IS,NX,NQ,4)
          CALL USWFM(1HM,1,M,IS,NL,NX,4)
          CALL USWFM(2HDM,2,DM,IS,NL,NX,4)
          CALL USWFM(1HQ,1,Q,IS,NK,NK,4)
          CALL USWFM(1HW,1,W,IS,NQ,NQ,4)
C   FORMS R=ROEC*R
          CALL SCAMUL(ROEC,IS,R,IS,R,NM,NM)
C   FORMS V=ROEE*V
          CALL SCAMUL(ROEE,IS,V,IS,V,NL,NL)
C  FORMS WW=D*W*DT
          CALL MP32(IS,NX,D,IS,NQ,W,IS,NQ,NX,D,IS,WW)
          CALL MP31(IS,NX,C,IS,NK,Q,IS,NK,NX,C,IS,CQC)
```

```
      CALL USWFM(1HR,1,R,IS,NM,NM,4)
      CALL USWFM(1HV,1,V,IS,NL,NL,4)
19    CALL EIGRF(A,NX,IS,2,EIGA,EE,IB,WK,IER)
      CALL USWCM(4HEIGA,4,EIGA,IS,NX,1,4)
20    CALL SSLOG(IS,A,IS,B,IS,WW,IS,M,IS,V,IS,CQC,IS,R,NX,NM,NL
     1,0,.FALSE.,WK,IS,S,IS,K,IS,G,IS,P,IS,F,EIG,EE,RJ,RJX,RJU)
      CALL USWFM(1HG,1,G,IS,NM,NX,4)
      CALL USWFM(1HF,1,F,IS,NX,NM,4)
      CALL USWFM(1HK,1,K,IS,NX,NX,4)
      CALL USWFM(1HP,1,P,IS,NX,NX,4)
22    CALL MULT(B,G,BG,NX,NM,NX,IS,IS,IS)
23    CALL MULT(F,M,FM,NX,NL,NX,IS,IS,IS)
      CALL MATADD(A,BG,ABG,NX,NX,IS)
      CALL MATSUB(A,FM,AFM,NX,IS,IS,IS)
      CALL EIGRF (ABG,NX,IS,2,EIGN,EE,IB,WK,IER)
39    CALL USWCM (14HEIGENS OF A+BG,14,EIGN,IB,NX,1,4)
C
      CALL EIGRF(AFM,NX,IS,2,EIGN,EE,IB,WK,IER)
C
40    CALL USWCM (14HEIGENS OF A-FM,14,EIGN,IB,NX,1,4)
24    CALL MADDSB(A,BG,FM,NX,AC,IS,IS,IS,IS)
25    CALL MATFT2(A,BG,FM,AC,NX,NX,NX,NX,ACL,2*NX
     1,2*NX,IB,IS,IS)
      CALL USWFM(3HACL,3,ACL,IB,2*NX,2*NX,4)
      NACL=2*NX
      NECL=NACL
30    CALL ABS(G,NM,NX,IS)
31    CALL ABS(F,NX,NL,IS)
32    CALL MULT(DB,G,DBG,NX,NM,NX,IS,IS,IS)
33    CALL MULT(F,DM,FDM,NX,NL,NX,IS,IS,IS)
34    CALL MATFT2(DA,DBG,FDM,ZERO,NX,NX,NX,NX,ECL,NECL
     1,NECL,IB,IS,IS)
      CALL USWFM(1HE,1,ECL,IB,NECL,NECL,4)
      WRITE(36,51)((ECL(I,J),J=1,NECL),I=1,NECL)
      CALL MATADD(ECL,ACL,ACL,NECL,NECL,IB)
      CALL USWFM(5HACL+E,5,ACL,IB,NECL,NECL,4)
      CALL EIGRF(ACL,NECL,IB,2,EIGACL,EE,IB,WK,IER)
      CALL USWCM(14HEIGEN OF ACL+E,14,EIGACL,IB,NECL,1,4)
      STOP
      END
```

```
C     PROGRAM TRUNK
C THIS PROGRAM FORMS CLOSED LOOP MATRIX AND PETURBATION MATRIX
C FOR A TRUNCATED MODES PROBLEM
      REAL AR(14,14),BR(14,14),CR(14,14),DR(14,14),MR(14,14),AT(14,14)
      REAL BT(14,14),CT(14,14),AD(28,28),WW(14,14),A(28,28)
      REAL DT(14,14),MT(14,14),Q(14,14),R(14,14),CQC(14,14)
      REAL W(14,14),V(14,14),S(14,14),ART(14,14),ATR(14,14)
      REAL K(14,14),P(14,14),FR(14,14),GR(14,14),WK(500),ZERO(14,14)
      REAL FRMR(14,14),BRGR(14,14),AC(14,14),RAFM(14,14),RABG(14,14)
      REAL ACL(28,28),AE(28,28),BTGR(14,14)
      REAL FRMT(14,14),ECL(28,28),EN(28,28)
      REAL CC(28,28),DDD(28,28),ENS(28,28)
C
      COMPLEX EIGA(28),EIG(14),EE(28,28),EI(28,28)
      COMPLEX EIGACL(28),EIGN(28)
C
       IS=14
       IB=28
       NXR=2
       NXT=1
       NX=NXR+NXT
       NM=1
       NQ=1
       NK=3
       NL=1
C
C     GIVE VALUES OF AR,ART,ATR AND AT
2      READ (15,*)((AR(I,J),J=1,NXR),I=1,NXR)
3      READ (15,*)((AT(I,J),J=1,NXT),I=1,NXT)
       READ(15,*)((ART(I,J),J=1,NXT),I=1,NXR)
       READ(15,*)((ATR(I,J),J=1,NXR),I=1,NXT)
C
C
C
C     GIVE VALUES OF BR&BT
5      READ (15,*)((BR(I,J),J=1,NM),I=1,NXR)
6      READ (15,*)((BT(I,J),J=1,NM),I=1,NXT)
C
C     GIVE VALUES OF CR&CT
8      READ(15,*)((CR(I,J),J=1,NXR),I=1,NK)
9      READ(15,*)((CT(I,J),J=1,NXT),I=1,NK)
```

```
C
C
C     GIVE VALUES OF MR&MT
14      READ (15,*)((MR(I,J),J=1,NXR),I=1,NL)
15      READ (15,*)((MT(I,J),J=1,NXT),I=1,NL)
C
C     GIVE VALUES OF Q(NK,NK)
C     Q IS SYM,+VE DEFN MATRIX
16      READ(15,*)((Q(I,J),J=1,NK),I=1,NK)
C
C     GIVE VALUES OF ROEC
        READ(15,*)ROEC
        DO  101 I=1,NM
        DO  101 J=1,NM
        R(I,J)=0
101     R(I,I)=1


C    GIVE VALUES OF ROEE
       READ(15,*)ROEE
        DO 102 I=1,NL
        DO 102 J=1,NL
        V(I,J)=0
102     V(I,I)=1
C
C GIVE VALUES OF W(NQ,NQ)
        READ(15,*)((W(I,J),J=1,NQ),I=1,NQ)
C
C CALLS IMSL ROUTINE FOR SETTING OUTPUT IDENTIFIER
        CALL UGETIO(3,0,35)
        CALL USWFM(2HAR,2,AR,IS,NXR,NXR,4)
        CALL USWFM(2HAT,2,AT,IS,NXT,NXT,4)
        CALL USWFM(3HART,3,ART,IS,NXR,NXT,4)
        CALL USWFM(3HATR,3,ATR,IS,NXT,NXR,4)
        CALL USWFM(2HBR,2,BR,IS,NXR,NM,4)
        CALL USWFM(2HBT,2,BT,IS,NXR,NM,4)
        CALL USWFM(2HCR,2,CR,IS,NK,NXR,4)
        CALL USWFM(2HCT,2,CT,IS,NK,NXT,4)
        CALL USWFM(2HDR,2,DR,IS,NXR,NQ,4)
        CALL USWFM(2HDT,2,DT,IS,NXT,NQ,4)
        CALL USWFM(2HMR,2,MR,IS,NL,NXR,4)
        CALL USWFM(2HMT,2,MT,IS,NL,NXT,4)
C   FORMS R=ROEC*R
        CALL SCAMUL(ROEC,IS,R,IS,R,NM,NM)
C   FORMS V=ROEE*V
        CALL SCAMUL(ROEE,IS,V,IS,V,NL,NL)
        CALL USWFM(1HQ,1,Q,IS,NK,NK,4)
        CALL USWFM(1HW,1,W,IS,NQ,NQ,4)
C  FORMS WW=D*W*DT
        CALL MP32(IS,NXR,DR,IS,NQ,W,IS,NQ,NXR,DR,IS,WW)
        CALL MP31(IS,NXR,CR,IS,NK,Q,IS,NK,NXR,CR,IS,CQC)
        CALL USWFM(1HR,1,R,IS,NM,NM,4)
        CALL USWFM(1HV,1,V,IS,NL,NL,4)
        CALL MATFT2(AR,ART,ATR,AT,NXR,NXT,NXR,NXT,
     1  A,NX,NX,IB,IS,IS)
19      CALL EIGRF(A,NX,IB,2,EIGA,EE,IB,WK,IER)
        CALL USWCM(4HEIGA,4,EIGA,IB,NX,1,4)
20      CALL SSLQG(IS,AR,IS,BR,IS,WW,IS,MR,IS,V,IS,CQC,IS,R,NXR,NM,NL
     1,0,.FALSE.,WK,IS,S,IS,K,IS,GF,IS,P,IS,FR,EIG,EE,J,JX,JU)
        CALL USWFM(2HGR,2,GR,IS,NM,NXR,4)
        CALL USWFM(2HFR,2,FR,IS,NXR,NM,4)
        CALL USWFM(1HK,1,K,IS,NXR,NXR,4)
        CALL USWFM(1HP,1,P,IS,NXR,NXR,4)
22      CALL MULT(BR,GR,BRGR,NXR,NM,NXR,IS,IS,IS)
23      CALL MULT(FR,MR,FRMR,NXR,NL,NXR,IS,IS,IS)
        CALL MULT(BT,GR,BTGR,NXT,NM,NXR,IS,IS,IS)
        CALL MULT(FR,MT,FRMT,NXR,NL,NXT,IS,IS,IS)
        CALL MATADD(AR,BRGR,RABG,NXR,NXR,IS)
        CALL MATSUB(AR,FRMR,RAFM,NXR,IS,IS,IS)
        CALL EIGRF (RABG,NXR,IS,2,EIGN,EE,IB,WK,IER)
39      CALL USWCM (17HEIGENS OF AR+BRGR,17,EIGN,IB,NXR,1,4)
```

```
         CALL EIGRF(RAFM,NXR,IS,2,EIGN,EE,IB,WK,IER)
C
40       CALL USWCM (17HEIGENS OF AR-FRMR,17,EIGN,IB,NXR,1,4)
24       CALL MADDSB(AR,BRGR,FRMR,NXR,AC,IS,IS,IS,IS)
C
        NACL=2*NXR+NXT
        NECL=NACL
C         CALL MATFT3
25         CALL MATFIT(AR,BRGR,ZERO,FRMR,AC,ZERO,ZERO,ZERO,AT,
     1  NXR,NXR,NXT,NXR,NXR,NXT,NACL,NACL,ACL,IS,IS,IS,IB)
         CALL USWFM(3HACL,3,ACL,IB,2*NXR+NXT,2*NXR+NXT,4)
        WRITE(36,51)((ACL(I,J),J=1,NACL),I=1,NACL)
C FORM E MATRIX
        SL=2
29          CALL SCAMUL(SL,IS,BRGR,IS,BRGR,NXR,NXR)
30          CALL SCAMUL(SL,IS,FRMR,IS,FRMR,NXR,NXR)
31          CALL SCAMUL(SL,IS,AC,IS,AC,NXR,NXR)
32          CALL ABS(BRGR,NXR,NXR,IS)
33          CALL ABS(FRMR,NXR,NXR,IS)
34          CALL ABS(AC,NXR,NXR,IS)
37          CALL ABS(ART,NXR,NXT,IS)
38          CALL ABS(ATR,NXT,NXR,IS)
41          CALL ABS(FRMT,NXR,NXT,IS)
42          CALL ABS(BTGR,NXT,NXR,IS)
43          CALL MATFIT(ZERO,BRGR,ART,FRMR,AC,FRMT,ATR,BTGR,ZERO,
     1    NXR,NXR,NXT,NXR,NXR,NXT,NECL,NECL,ECL,IS,IS,IS,IB)
C
      STOP
      END


C     PROGRAM COST:
C     THIS PROGRAM COMPUTES THE REGULATION COST,CONTROL COST AND TOTAL
C     COST FOR THE OPEN LOOP,NOMINAL CLOSED LOOP AND PERTURBED CLOSED
C     LOOP CASES AND THE PERFORMANCE INDEX.
C     DIMENSIONS:
      REAL A(9,9),DA(9,9),B(9,9),DB(9,9),W(9,9),M(9,9),DM(9,9),V(9,9)
      REAL Q(9,9),R(9,9),SS(9,9),KK(9,9),G(9,9),P(9,9),F(9,9),A12(9,9)
      REAL A21(9,9),A22(9,9),A32(9,9),A33(9,9),A34(9,9),A41(9,9)
      REAL A43(9,9),AF11(18,18),AF12(18,18),AF21(18,18),AF22(18,18)
      REAL D(9,9),DD(9,9),DF1(18,18),DF2(18,18),C(9,9),DC(9,9),C23(9,9)
      REAL CF(36,36),QF(36,36),WF(18,18),PR1(18,18),PR21(18,18)
      REAL PR22(18,18),PR2(18,18),PR31(18,18),PR32(18,18),PR33(18,18)
      REAL PR3(18,18),PF11(18,18),PF12(18,18),PF22(18,18),PF(36,36)
      REAL TPF12(18,18),WK(500),ZERO(9,9),WW(9,9),CQC(36,36),DUM(18,18)
      REAL POP(9,9)
      REAL JOP,JXN,JUN,JN,JXP,JUP,JP,JXNN,JUNN,JNN
      COMPLEX EIG1(36),EIG2(36),MOD1(36,36)
      COMPLEX IMOD1(36,36),MOD2(36,36),IMOD2(36,36),WKC(1500)
      IR=9
      IR2=18
2     IR4=36
      NN=1
      NM=1
      NQ=1
      NK=1
      NL=1
      NN2=2*NN
      NN4=4*NN
      NQL=NQ+NL
3     NKM2=2*(NK+NM)
C     INPUT VALUES:
8     READ(50,*)((A(I,J),J=1,NN),I=1,NN)
      READ(50,*)((DA(I,J),J=1,NN),I=1,NN)
71    READ(50,*)((B(I,J),J=1,NM),I=1,NN)
      READ(50,*)((DB(I,J),J=1,NM),I=1,NN)
72    READ(50,*)((C(I,J),J=1,NN),I=1,NK)
      READ(50,*)((DC(I,J),J=1,NN),I=1,NK)
73    READ(50,*)((D(I,J),J=1,NQ),I=1,NN)
      READ(50,*)((DD(I,J),J=1,NQ),I=1,NN)
74    READ(50,*)((M(I,J),J=1,NN),I=1,NL)
      READ(50,*)((DM(I,J),J=1,NN),I=1,NL)
76    READ(50,*)((W(I,J),J=1,NQ),I=1,NQ)
77    READ(50,*)((V(I,J),J=1,NL),I=1,NL)
      READ(50,*) ROEC
      READ(50,*) ROEE
      READ(50,*)((Q(I,J),J=1,NK),I=1,NK)
      READ(50,*) ITTY
81    CALL IDENT(R,NM,IR)
82    CALL IDENT(V,NL,IR)
83    CALL SCAMUL(ROEC,IR,R,IR,R,NM,NM)
84    CALL SCAMUL(ROEE,IR,V,IR,V,NL,NL)
C     OPEN LOOP COST
```

```
C
501   CALL MP32(IR,NN,D,IR,NQ,W,IR,NQ,NN,D,IR,WW)
502   CALL MEQ(A,DUM,NN,NN,IR,IR2)
503   CALL EIGRF(DUM,NN,IR2,1,EIG2,MOD1,IR4,WK,IER)
504   CALL CINV(MOD1,IMOD1,WK,IR4,NN)
505   CALL LYAP2(EIG2,IR4,MOD1,IMOD1,IR,WW,IR,POP,NN,WKC)
506   CALL MP31(IR,NN,C,IR,NK,Q,IR,NK,NN,C,IR2,DUM)
      JOP=TRAC2(IR,POP,IR2,DUM,NN,NN)


      CALL MP31(IR,NN,C,IR,NK,Q,IR,NK,NN,C,IR4,COC)
5     CALL SSLOG(IR,A,IR,B,IR,WW,IR,M,IR,V,IR4,COC,IR,R,NN,NM,NL,O
     1,.FALSE.,WK,IR,SS,IR,KK,IR,G,IR,P,IR,F,EIG1,WKC,JN,JXN,JUN)
      CALL UGETIO(3,0,ITTY)
507   WRITE(ITTY,*)('INPUT VALUES')
C      CALL UGETIO(1,03,33)
      CALL USWFM(4HROEE,4,ROEE,1,1,1,4)
      CALL USWFM(4HROEC,4,ROEC,1,1,1,4)
      CALL USWFM(8HMATRIX A,8,A,IR,NN,NN,4)
      CALL USWFM(9HMATRIX DA,9,DA,IR,NN,NN,4)
      CALL USWFM(8HMATRIX B,8,B,IR,NN,NM,4)
508   CALL USWFM(9HMATRIX DB,9,DB,IR,NN,NM,4)
      CALL USWFM(8HMATRIX C,8,C,IR,NK,NN,4)
      CALL USWFM(9HMATRIX DC,9,DC,IR,NK,NN,4)
      CALL USWFM(8HMATRIX D,8,D,IR,NN,NQ,4)
      CALL USWFM(9HMATRIX DD,9,DD,IR,NN,NQ,4)
      CALL USWFM(8HMATRIX M,8,M,IR,NL,NN,4)
      CALL USWFM(9HMATRIX DM,9,DM,IR,NL,NN,4)
      CALL USWFM(8HMATRIX W,8,W,IR,NQ,NQ,4)
      CALL USWFM(8HMATRIX V,8,V,IR,NL,NL,4)
      CALL USWFM(8HMATRIX Q,8,Q,IR,NK,NK,4)
      CALL USWFM(8HMATRIX R,8,R,IR,NM,NM,4)
      WRITE(ITTY,*)('OUTPUT VALUES')
      CALL USWFM(8HMATRIX G,8,G,IR,NM,NN,4)
      CALL USWFM(8HMATRIX F,8,F,IR,NN,NL,4)
      CALL USWCM(16HEIGEN VALUES/OPL,16,EIG2,IR4,NN,1,4)
      CALL USWCM(16HEIGEN VALUES/NCL,16,EIG1,IR4,NN2,1,4)
C
C      NOMINAL AND PERTURBED CLOSED LOOP COSTS
C
10    CALL MULT (B,G,A12,NN,NM,NN,IR,IR,IR)
15    CALL MULT(F,M,A21,NN,NL,NN,IR,IR,IR)
20    CALL MADDSB(A,A12,A21,NN,A22,IR,IR,IR,IR)
25    CALL MULT(DB,G,A32,NN,NM,NN,IR,IR,IR)
30    CALL MATADD(A,DA,A33,NN,NN,IR)
35    CALL MATADD(A12,A32,A34,NN,NN,IR)
40    CALL MULT(F,DM,A41,NN,NL,NN,IR,IR,IR)
45    CALL MATADD(A21,A41,A43,NN,NN,IR)
      CALL MATFT2(A,A12,A21,A22,NN,NN,NN,NN,AF11,NN2,NN2,IR2,IR,IR)
      CALL MATFT2(DA,A32,A41,ZERO,NN,NN,NN,NN,AF21,NN2,NN2,IR2,IR,IR)
      CALL MATFT2(A33,A34,A43,A22,NN,NN,NN,NN,AF22,NN2,NN2,IR2,IR,IR)
52    CALL MATFT2(D,ZERO,ZERO,F,NN,NN,NQ,NL,DF1,NN2,NQL
     1,IR2,IR,IR)
60    CALLMATFT2(DD,ZERO,ZERO,ZERO,NN,NN,NQ,NL,DF2,NN2
     1,NQL,IR2,IR,IR)
      WRITE(5,*)((AF11(I,J),J=1,NN2),I=1,NN2)
      WRITE(5,*)((AF21(I,J),J=1,NN2),I=1,NN2)
65    CALL MATADD(C,DC,C23,NK,NN,IR)
70    CALL MATFT4(C,ZERO,ZERO,ZERO,DC,ZERO,C23,ZERO,ZERO,G,ZERO,ZERO
     1,ZERO,ZERO,ZERO,G,CF,NK,NK,NM,NM,NN,NN,NN,NN,NKM2,NN4,IR,IR
     1,IR,IR,IR4)
90    CALL MATFT2(W,ZERO,ZERO,V,NQ,NL,NQ,NQ,WF,NQL,NQL,IR2,IR,IR)
C     SOLVE LIAPUNOV'S EQUATION:
C     (A)  PF11.AF11T+AF11.PF11+DF11.W.DF11T=0
95    CALL MP32(IR2,NN2,DF1,IR2,NQL,WF,IR2,NQL,NN2,DF1
     1,IR2,PR1)
      CALL MEQ(AF11,DUM,NN2,NN2,IR2,IR2)
100   CALL EIGRF(DUM,NN2,IR2,1,EIG1,MOD1,IR4,WK,IER)
105   CALL CINV(MOD1,IMOD1,WK,IR4,NN2)
110   CALL LYAP2(EIG1,IR4,MOD1,IMOD1,IR2,PR1,IR2,PF11
     1,NN2,WKC)
```

```
C      (B)  PF12.AF22T+AF11.PF12+PF11.AF21T+DF1.W.DF2T=0
115    CALL MULRRT(PF11,AF21,PR21,NN2,NN2,NN2,IR2,IR2,IR2)
120    CALL MP32 (IR2,NN2,DF1,IR2,NQL,WF,IR2,NQL,NN2,DF2,IR2
      1,PR22)
125    CALL MATADD(PR21,PR22,PR2,NN2,NN2,IR2)
       CALL MEQ(AF22,DUM,NN2,NN2,IR2,IR2)
130    CALL EIGRF(DUM,NN2,IR2,1,EIG2,MOD2,IR4,WK,IER)
135    CALL CINV(MOD2,IMOD2,WK,IR4,NN2)
140    CALL LYAP5(NN2,EIG1,IR4,MOD1,IMOD1,NN2,EIG2,IR4,MOD2
      1,IMOD2,IR2,PR2,IR2,PF12,WKC)
       CALL USWCM(16HEIGEN VALUES/PCL,16,EIG1,IR4,NN2,1,4)
C      (C)  PF22.AF22T+AF22.PF22+TPF12.AF21T+AF21.PF12
C           +DF2.W.DF2T = 0
       DO 145 I=1,NN2
       DO 145 J=1,NN2
145    TPF12(I,J)=PF12(J,I)
150    CALL MULRRT(TPF12,AF21,PR31,NN2,NN2,NN2,IR2,IR2,IR2)
155    CALL MULT(AF21,PF12,PR32,NN2,NN2,NN2,IR2,IR2,IR2)
160    CALL MP32(IR2,NN2,DF2,IR2,NQL,WF,IR2,NQL,NN2,DF2,IR2
      1,PR33)
165    CALL MATADD(PR31,PR32,DUM,NN2,NN2,IR2)
170    CALL MATADD(DUM,PR33,PR3,NN2,NN2,IR2)
185    CALL LYAP2(EIG2,IR4,MOD2,IMOD2,IR2,PR3,IR2,PF22,NN2,WKC)
C      FORM MAT PF:
195    CALL MATFT2(PF11,PF12,TPF12,PF22,NN2,NN2,NN2,NN2,PF,NN4
      1,NN4,IR4,IR2,IR2)
C      FORM CFT.QF1.CF, CFT.QF2.CF, CFT.QF.CF
196    CALL MATFT4(Q,Q,ZERO,ZERO,Q,Q,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
      1,ZERO,ZERO,QF,NK,NK,NM,NM,NK,NK,NM,NM,NKM2,NKM2,IR,IR,IR,IR,IR4)
200    CALL MP31(IR4,NN4,CF,IR4,NKM2,QF,IR4,NKM2,NN4,CF,IR4,CQC)
201    JXP=TRAC2(IR4,PF,IR4,CQC,NN4,NN4)
202    CALL MATFT4(ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
      1,R,R,ZERO,ZERO,R,R,QF,NK,NK,NM,NM,NK,NK,NM,NM,NKM2,NKM2,IR,IR
      1,IR,IR,IR4)
205    CALL MP31(IR4,NN4,CF,IR4,NKM2,QF,IR4,NKM2,NN4,CF,IR4,CQC)
206    JUP=TRAC2(IR4,PF,IR4,CQC,NN4,NN4)
207    CALL MATFT4(Q,Q,ZERO,ZERO,Q,Q,ZERO,ZERO,ZERO,ZERO,R,R,ZERO,ZERO,
      1R,R,QF,NK,NK,NM,NM,NK,NK,NM,NM,NKM2,NKM2,IR,IR,IR,IR,IR4)
210    CALL MP31(IR4,NN4,CF,IR4,NKM2,QF,IR4,NKM2,NN4,CF,IR4,CQC)
225    JP=TRAC2(IR4,PF,IR4,CQC,NN4,NN4)
       CALL MATFT4(Q,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
      1,R,ZERO,ZERO,ZERO,ZERO,ZERO,QF,NK,NK,NM,NM,NK,NK,NM,NM,NKM2
      1,NKM2,IR,IR,IR,IR,IR4)
1001   CALL MP31(IR4,NN4,CF,IR4,NKM2,QF,IR4,NKM2,NN4,CF,IR4,CQC)
1002   JNN=TRAC2(IR4,PF,IR4,CQC,NN4,NN4)
1003   CALL MATFT4(Q,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
      1,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,QF,NK,NK,NM,NM,NK,NK,NM,NM
      1,NKM2,NKM2,IR,IR,IR,IR,IR4)
1004   CALL MP31(IR4,NN4,CF,IR4,NKM2,QF,IR4,NKM2,NN4,CF,IR4,CQC)
1005   JXNN=TRAC2(IR4,PF,IR4,CQC,NN4,NN4)
1006   CALL MATFT4(ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
      1,ZERO,R,ZERO,ZERO,ZERO,ZERO,ZERO,QF,NK,NK,NM,NM,NK,NK,NM,
      1NM,NKM2,NKM2,IR,IR,IR,IR,IR4)
1007   CALL MP31(IR4,NN4,CF,IR4,NKM2,QF,IR4,NKM2,NN4,CF,IR4,CQC)
       JUNN=TRAC2(IR4,PF,IR4,CQC,NN4,NN4)
       BETAPR=ABS(JXP-JXNN)/JXNN
       CALL IDENT(Q,NK,IR)
1008   CALL MATFT4(Q,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
      1,ZERO,ZERO,ZERO,ZERO,ZERO,QF,NK,NK,NM,NM,NK,NK,NM,NM,NKM2,NKM2,
      1IR,IR,IR,IR,IR4)
       CALL MP31(IR4,NN4,CF,IR4,NKM2,QF,IR4,NKM2,NN4,CF,IR4,CQC)
1009   YTY=TRAC2(IR4,PF,IR4,CQC,NN4,NN4)
1010   CALL MATFT4(ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
      1,ZERO,Q,ZERO,ZERO,ZERO,ZERO,ZERO,QF,NK,NK,NM,NM,NK,NK,NM,NM
      1,NKM2,NKM2,IR,IR,IR,IR,IR4)
1011   CALL MP31(IR4,NN4,CF,IR4,NKM2,QF,IR4,NKM2,NN4,CF,IR4,CQC)
       UTU=TRAC2(IR4,PF,IR4,CQC,NN4,NN4)
       WRITE(ITTY,*)('OPEN LOOP COST')
       CALL USWFM(14HOPEN LOOP COST,14,JOP,1,1,1,4)
       WRITE(ITTY,*)('NOM.CLOSED LOOP COSTS')
       CALL USWFM(15HREGULATION COST,15,JXNN,1,1,1,4)
       CALL USWFM(12HCONTROL COST,12,JUNN,1,1,1,4)
       CALL USWFM(10HTOTAL COST,10,JNN,1,1,1,4)
       WRITE(ITTY,*)('PERT.CLOSED LOOP COSTS')
       CALL USWFM(15HREGULATION COST,15,JXP,1,1,1,4)
       CALL USWFM(12HCONTROL COST,12,JUP,1,1,1,4)
       CALL USWFM(10HTOTAL COST,10,JP,1,1,1,4)
       CALL USWFM(5HY T Y,5,YTY,1,1,1,4)
       CALL USWFM(5HU T U,5,UTU,1,1,1,4)
       YTY1=SQRT(YTY)
       UTU1=SQRT(UTU)
       CALL USWFM(8HYTY**1/2,8,YTY1,1,1,1,4)
       CALL USWFM(8HUTU**1/2,8,UTU1,1,1,1,4)
       WRITE(ITTY,*)('PERF.ROB.INDEX')
       CALL USWFM(7HBETA PR,7,BETAPR,1,1,1,4)
12     FORMAT(F18.5)
13     FORMAT(3F18.5)
       STOP
```

```
      SUBROUTINE MADDSB (A,B,C,N,ABC,IA,IB,IC,IABC)
C*************************************************************
C  THIS DOES  ABC=A+B-C
      REAL A(IA,N),B(IB,N),C(IC,N),ABC(IABC,N)
      DO 41 I = 1,N
      DO 41 J =1, N
41    ABC(I,J)=A(I,J)+B(I,J)-C(I,J)
      RETURN
      END


      SUBROUTINE MATFIT (A11,A12,A13,A21,A22,A23,A31,A32,A33
     1 ,NR1,NR2,NR3,NC1,NC2,NC3,NR,NC,A,IR1,IR2,IR3,IR)
      REAL A11(IR1,NC1),A12(IR1,NC2),A13(IR1,NC3),
     1  A21(IR2,NC1),A22(IR2,NC2),
     1  A23(IR2,NC3),A31(IR3,NC1),A32(IR3,NC2),A33(IR3,NC3),A(IR,NC)
      DO 40 I=1,NR
      DO 40 J=1,NC
40    A(I,J)=0
C     THIS SUBR COMPILES MATRICES INTO A ONE MATRIX
      DO 41 I =1,NR1
      DO 41 J =1,NC1
      A(I,J) = A11 (I,J)
41    CONTINUE
      DO 42 I =1,NR1
      DO 42 J =1,NC2
      A(I,NC1+J)=A12(I,J)
42    CONTINUE
      DO 43 I =1,NR1
      DO 43 J =1,NC3
43    A(I,NC1+NC2+J)=A13(I,J)
      DO 44 I =1,NR2
      DO 44 J =1,NC1
44    A(NR1+I,J) = A21(I,J)
      DO 45 I =1,NR2
      DO 45 J =1,NC2
45    A(NR1+I,NC1+J)= A22(I,J)
      DO 46 I =1,NR2
      DO 46 J =1,NC3
46    A(NR1+I,NC1+NC2+J)=A23(I,J)
      DO 47 I =1,NR3
      DO 47 J =1,NC1
47    A(NR1+NR2+I,J)=A31(I,J)
      DO 48 I = 1,NR3
      DO 48 J =1, NC2
48    A(NR1+NR2+I,NC1+J)=A32(I,J)
      DO 49 I = 1,NR3
      DO 49 J =1, NC3
49    A(NR1+NR2+I,NC1+NC2+J)=A33(I,J)
      RETURN
      END


      SUBROUTINE ABS (AA,NR,NC,IA)
C*************************************************************
C THIS GIVES ABSOLUTE VALUES OF ALL ELEMENTS OF
C AA MATRIX AND OUTPUT IS STORED BACK INTO 'AA'
      REAL AA(IA,NC)
      DO 71 I=1,NR
      DO 71 J=1,NC
71    AA(I,J)=ABS(AA(I,J))
      RETURN
      END
      SUBROUTINE DIAMAT(AD,ND,EIG,IAD,IEIG)
C*************************************************************
C THIS SUBROUTINE MAKES 'AD' A DIAGONAL MATRIX
C DIAGONAL ELEMENTS AS REAL PART EIGEN VALUES OF
C ANOTHER MATRIX OF SIMILAR ORDER
      REAL AD(IAD,ND)
      COMPLEX EIG(IEIG)
      DO 81 I=1,ND
      DO 81 J=1,ND
81    AD(I,J)=0
      DO 82 I=1,ND
82    AD(I,I) =REAL(EIG(I))
      RETURN
      END
```

```
C
C       SUBROUTINE MATFT4
C
C ******************************************************************:
C ***
C *** THIS SUBROUTINE COMPILES  4*4 MATRICES INTO A SINGLE MATRIX.
C ***
C ***     A11  A12  A13  A14
C ***     A21  A22  A23  A24       =A
C ***     A31  A32  A33  A34
C ***     A41  A42  A43  A44
C ***
C ***   A11-NR1*NC1, A12-NR1*NC2, A13-NR1*NC3, A14-NR1*NC4
C ***   A21-NR2*NC1, A22-NR2*NC2, A23-NR2*NC3, A24-NR2*NC4
C ***   A31-NR3*NC1, A32-NR3*NC2, A33-NR3*NC3, A34-NR3*NC4
C ***   A41-NR4*NC1, A42-NR4*NC2, A43-NR4*NC3, A44-NR4*NC4
C ***                        A-NR*NC
C ***
C ***   MR1,MR2,MR3,MR4,MC1,MC2,MC3,MC4,MR,MC-DIMENSIONS
C ***
C ******************************************************************
C
      SUBROUTINE MATFT4(A11,A12,A13,A14,A21,A22,A23,A24,A31,A32
     1,A33,A34,A41,A42,A43,A44,A,NR1,NR2,NR3,NR4,NC1,NC2,NC3,NC4
     1,NR,NC,MR1,MR2,MR3,MR4,MR)
C
      REAL A11(MR1,MR1),A12(MR1,MR2),A13(MR1,MR3),A14(MR1,MR4)
      REAL A21(MR2,MR1),A22(MR2,MR2),A23(MR2,MR3),A24(MR2,MR4)
      REAL A31(MR3,MR1),A32(MR3,MR2),A33(MR3,MR3),A34(MR3,MR4)
      REAL A41(MR4,MR1),A42(MR4,MR2),A43(MR4,MR3),A44(MR4,MR4)
      REAL A(MR,MR)
C
      DO 50 I=1,NR
      DO 50 J=1,NC
50    A(I,J)=0
      DO 100 I=1,NR1
      DO 100 J=1,NC1
100   A(I,J)=A11(I,J)
C
      DO 110 I=1,NR1
      DO 110 J=1,NC2
110   A(I,NC1+J)=A12(I,J)
C
      DO 120 I=1,NR1
      DO 120 J=1,NC3
120   A(I,NC1+NC2+J)=A13(I,J)
C
      DO 130 I=1,NR1
      DO 130 J=1,NC4
130   A(I,NC1+NC2+NC3+J)=A14(I,J)
C
      DO 140 I=1,NR2
      DO 140 J=1,NC1
140   A(NR1+I,J)=A21(I,J)
C
      DO 150 I=1,NR2
      DO 150 J=1,NC2
150   A(NR1+I,NC1+J)=A22(I,J)
C
      DO 160 I=1,NR2
      DO 160 J=1,NC3
160   A(NR1+I,NC1+NC2+J)=A23(I,J)
C
      DO 170 I=1,NR2
      DO 170 J=1,NC4
170   A(NR1+I,NC1+NC2+NC3+J)=A24(I,J)
C
      DO 180 I=1,NR3
      DO 180 J=1,NC1
180   A(NR1+NR2+I,J)=A31(I,J)
C
      DO 190 I=1,NR3
      DO 190 J=1,NC2
190   A(NR1+NR2+I,NC1+J)=A32(I,J)
C
      DO 200 I=1,NR3
      DO 200 J=1,NC3
200   A(NR1+NR2+I,NC1+NC2+J)=A33(I,J)
C
      DO 210 I=1,NR3
      DO 210 J=1,NC4
210   A(NR1+NR2+I,NC1+NC2+NC3+J)=A34(I,J)
C
```

```
      DO 220 I=1,NR4
      DO 220 J=1,NC1
220   A(NR1+NR2+NR3+I,J)=A41(I,J)
C
      DO 230 I=1,NR4
      DO 230 J=1,NC2
230   A(NR1+NR2+NR3+I,NC1+J)=A42(I,J)
C
      DO 240 I=1,NR4
      DO 240 J=1,NC3
240   A(NR1+NR2+NR3+I,NC1+NC2+J)=A43(I,J)
C
      DO 250 I=1,NR4
      DO 250 J=1,NC4
250   A(NR1+NR2+NR3+I,NC1+NC2+NC3+J)=A44(I,J)
C
      RETURN
      END

        SUBROUTINE MATFT2 (A11,A12,A21,A22,NR1,NR2,
     1NC1,NC2,A,NR,NC,IR,IR1,IR2)
C*******************************************************************
C       THIS FORMS A MATRIX FROM A11,A12,A21,A22
C
C        A11  A12
C        A21  A22           TO FORM  A
C      NR=NR1+NR2; NC=NC1+NC2;
C
C         NR=NR1+NR2
C         NC=NC1+NC2
C
        REAL A(IR,IR),A11(IR1,IR1),A12(IR1,IR2)
        REAL A21(IR2,IR1),A22(IR2,IR2)
        DO 10 I=1,NR
        DO 10 J=1,NC
10      A(I,J)=0
        DO 1  I=1,NR1
        DO 1  J=1,NC1
1       A(I,J) =A11(I,J)
        DO 2 I=1,NR1
        DO 2 J=1,NC2
2       A(I,NC1+J)=A12(I,J)
        DO 3 I=1,NR2
        DO 3 J=1,NC1
3       A(NR1+I,J)=A21(I,J)
        DO 4 I=1,NR2
        DO 4 J=1,NC2
4       A(NR1+I,NC1+J)=A22(I,J)
        RETURN
        END

        SUBROUTINE STABLE(A,B,C,WK,NW,D,AE,N,W,Z
     1  ,IA,IB,IC,ID,IAE,IZ)
C*****************************************************************
C THIS SUB FINDS EIGEN VALUES OF
C    [ (E)S*(INV(AD))S ]SYM
        REAL A(IA,N),B(IB,N),C(IC,N),WK(NW),D(ID,N),AE(IAE,N)
        INTEGER N,IJOB,IER
        COMPLEX W(IZ),Z(IZ,N),ZN
        IJOB=0
        DO 55 I=1,N
        DO 55 J=1,N
55      AE(I,J)=(A(I,J)+A(J,I))/2
C FORMS INV DIAGONAL MATRIX ONLY
        DO 66 I=1,N
        DO 66 J=1,N
        IF(I.EQ.J) GO TO 64
        B(I,J)=B(I,J)
        GO TO 66
64      CONTINUE
        B(I,I)=1.0/B(I,I)
66      CONTINUE
CC        WRITE(5,*)('AD MATRIX')
C        WRITE(5,*)((B(I,J),J=1,N),I=1,N)
        CALL VMULFF (AE,B,N,N,N,IAE,IB,C,IC,IER)
C        WRITE(5,*)('AE MATR')
C        WRITE(5,*)((AE(I,J),J=1,N),I=1,N)
C        WRITE(5,*)('C MAT')
C        WRITE(5,*)((C(I,J),J=1,N),I=1,N)
        DO 9 I=1,N
        DO 9 J=1,N
9       D(I,J)= ((C(I,J)+C(J,I))/2
        CALL EIGRF (D,N,ID,IJOB,W,Z,IZ,WK,IER)
        END
```

```
      SUBROUTINE MARS(EIG,IR,N,SMALL)
C THIS SUBROUTINE FINDS MIN VALUE REAL PART OF COMPLEX
C VALUES
C       REAL REL(IR)
      COMPLEX EIG(IR)
      SMALL=REAL(EIG(1))
      SMALL=ABS(SMALL)
      DO 1 I=1,N
      REL=REAL(EIG(I))
      REL=ABS(REL)
      IF(REL.GE.SMALL) GO TO 1
      SMALL=REL
1     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MARB(EIG,IR,N,BIG)
      COMPLEX EIG(IR)
      BIGI=REAL(EIG(1))
      BIG=ABS(BIGI)
      DO 1 I=1,N
      REL=REAL(EIG(I))
      REL=ABS(REL)
      IF(REL.LE.BIG) GO TO 1
      BIG=REL
1     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STABR(ACL,ECL,ALPHA,ID,AE,EIG,Z,N,
     1IR,IZ,IEIG,WK,IWK)
      REAL ACL(IR,N),ECL(IR,N),ID(IR,N),AE(IR,N),WK(IWK)
      COMPLEX EIG(IEIG),Z(IZ,N),ZN
      IJOB=2
12      CALL IDENT(ID,N,IR)
13      CALL SCAMUL(ALPHA,IR,ID,IR,ID,N,N)
14      CALL MADD(ACL,ID,AE,N,N,IR,IR,IR,1)
15      CALL ABS(AE,N,N,IR)
16      CALL MADD(AE,ECL,AE,N,N,IR,IR,IR,1)
      DO 1 I=1,N
      DO 1 J=1,N
1     ID(I,J)=(AE(I,J)+AE(J,I))/2
      ALPHAI=1/ALPHA
17      CALL SCAMUL(ALPHAI,IR,ID,IR,ID,N,N)
18      CALL EIGRF(ID,N,IR,IJOB,EIG,Z,IZ,WK,IER)
      RETURN
      END
      SUBROUTINE MAR(EIG,IR,N,SMALL)
C THIS SUBROUTINE FINDS MIN VALUE REAL PART OF COMPLEX
C VALUES
C       REAL REL(IR)
      COMPLEX EIG(IR)
      SMALL=REAL(EIG(1))
      DO 1 I=1,N
      REL=REAL(EIG(I))
      IF(REL.GE.SMALL) GO TO 1
      SMALL=REL
1     CONTINUE
      END
```

```
      PROGRAM TEST
      REAL ACL(8,8),ECL(8,8),CC(8,8),AE(8,8),WK(1800)
      COMPLEX EIGN(8),EE(8,8)
      IB=8
      N=4
      WRITE(5,*)('GIVE SIZE OF MATRIX')
CD      READ(5,*)N
      WRITE(5,*)('INPUT VALUES OF ACL')
      READ(36,*)((ACL(I,J),J=1,N),I=1,N)
      WRITE(5,*)('INPUT VALUES OF ECL')
      READ(36,*)((ECL(I,J),J=1,N),I=1,N)
      DO 1 I=1,25
      WRITE(5,*)('GIVE VALUE OF ALPHA')
      READ(5,*)ALPHA
      CALL STABR(ACL,ECL,ALPHA,CC,AE,EIGN,EE,N,
     1IB,IB,IB,WK,1800)
      CALL UGETIO(3,0,5)
C      CALL USWFM(3HACL,3,ACL,IB,N,N,4)
C      CALL USWFM(3HECL,3,ECL,IB,N,N,4)
      CALL USWCM (6HEIGENS,6,EIGN,IB,N,1,4)
      CALL MARB(EIGN,IR,N,BIG)
      WRITE(5,*)('/MAX EIGEN/ = ',BIG)
      IF(BIG.LT.1)GO TO 10
      IF(BIG.GE.1)WRITE(5,*)('SYS IS NOT STABLE')
      WRITE(5,*)('TYPE 1 TO CONTINUE')
      WRITE(5,*)('TYPE 2 TO STOP')
      READ(5,*)NT
      IF(NT.EQ.1) GO TO 1
      GO TO 100
1     CONTINUE
10    CONTINUE
      WRITE(5,*)('SYS IS STABLE')
      CALL EIGRF(ACL,N,IB,2,EIGN,EE,IB,WK,IER)
      CALL USWCM(6HEIGACL,6,EIGN,IB,N,1,4)
      CALL MARS(EIGN,IB,N,SMACL)
      WRITE(5,*)('SMACL =',SMACL)
      CALL MATADD(ECL,ACL,ACL,N,N,IB)
      CALL EIGRF(ACL,N,IB,2,EIGN,EE,IB,WK,IER)
      CALL USWCM(6HACLECL,6,EIGN,IB,N,1,4)
      CALL MARS(EIGN,IB,N,SMPER)
      WRITE(5,*)('SMPER =',SMPER)
      BSTAB=(ABS(SMACL-SMPER))/SMACL
      WRITE(5,*)('BSTAB= ',BSTAB)
100   CONTINUE
      STOP
      END
```

Application Example:

Consider the Scalar System

$$\dot{x} = -x + u, \quad x_o = 1, \quad J_n = \int_o^\infty (x^2 + \rho_c u^2)\, dt$$

i.e. $a = -1.$, $b = 1$

Let $|\Delta a| = 0.5$; $|\Delta b| = 1.207$. The closed loop system is given by

$$\dot{x} = (a + bg)\, x = a_{CL}\, x$$

and the perturbation is given by Δa_{CL}.

We now consider three cases of the perturbation Δa_{CL}.

Case I: $\Delta a_{CL} = |\Delta a| + |\Delta b| g$

Case II: $\Delta a_{CL} = |\Delta a| + |\Delta b|\, |g|$

Case III: $\Delta a_{CL} = -(|\Delta a| + |\Delta b|\, |g|)$

The robustness indices $\beta_{S.R.}$ and $\beta_{P.R.}$ are calculated for different values of ρ_c and tabulated in Table I. Note that all the controllers corresponding to the range of ρ_c considered satisfy the stability robustness condition.

Note that the testing of stability condition is done with the perturbation Δa_{CL} corresponding to Case II to satisfy theorem B. Then by virtue of theorem B, if the perturbed system with Case II perturbation is stable then the perturbed systems of Cases I & III will also be stable.

ρ_c	$(x^Tx)^{\frac{1}{2}}$	$u^Tu^{\frac{1}{2}}$	Case I $\beta_{S.R.}$	Case I $\beta_{P.R.}$	Case II $\beta_{S.R.}$	Case II $\beta_{P.R.}$	Case III $\beta_{S.R.}$	Case III $\beta_{P.R.}$
1/8	0.408	0.816	0.638	0.39	0.97	0.494	0.97	0.494
1/4	0.473	0.584	0.443	0.31	0.89	9.15	0.89	0.47
1/2	0.537	0.393	0.22	0.18	0.798	3.97	0.798	0.44
3/4	0.572	0.302	0.08	0.08	0.744	2.91	0.744	0.43
1	0.594	0.246	0	0	0.707	2.4	0.707	0.414
5/4	0.610	0.209	0.065	0.07	0.680	2.125	0.680	0.405
3/2	0.6225	0.181	0.116	0.132	0.658	1.93	0.658	0.397
2	0.639	0.143	0.187	0.23	0.63	1.70	0.63	0.385

From the above table, it can be observed that Case II gives the worst case $\beta_{S.R.}$ and worst case $\beta_{P.R.}$. Thus, we use these indices for comparison of different controllers from stability and performance point of views. The information corresponding to Case II is presented graphically in figures A,B,C and D. A reasonable choice for robust controller could be the one corresponding to ρ_c = 1/2, 3/4 or 1.

Since this is a scalar example it is easy to see which case of perturbation represents the worst case situation. This may not be easily inferred for matrix cases. To determine which case of perturbation (among the possible three cases considered) represents the worst case situation for general matrix case is suggested as a future research topic.

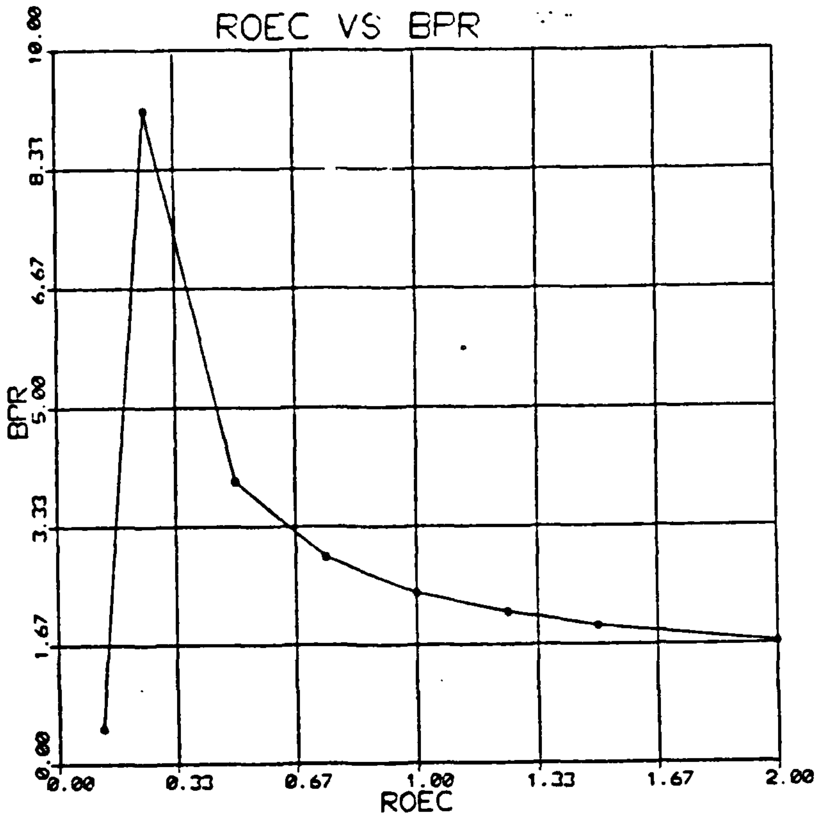

ROEC VS BPR
BPR
10.00
8.33
6.67
5.00
3.33
1.67
0.00
0.00
0.33
0.67
1.00
1.33
1.67
2.00
ROEC

Fig. A

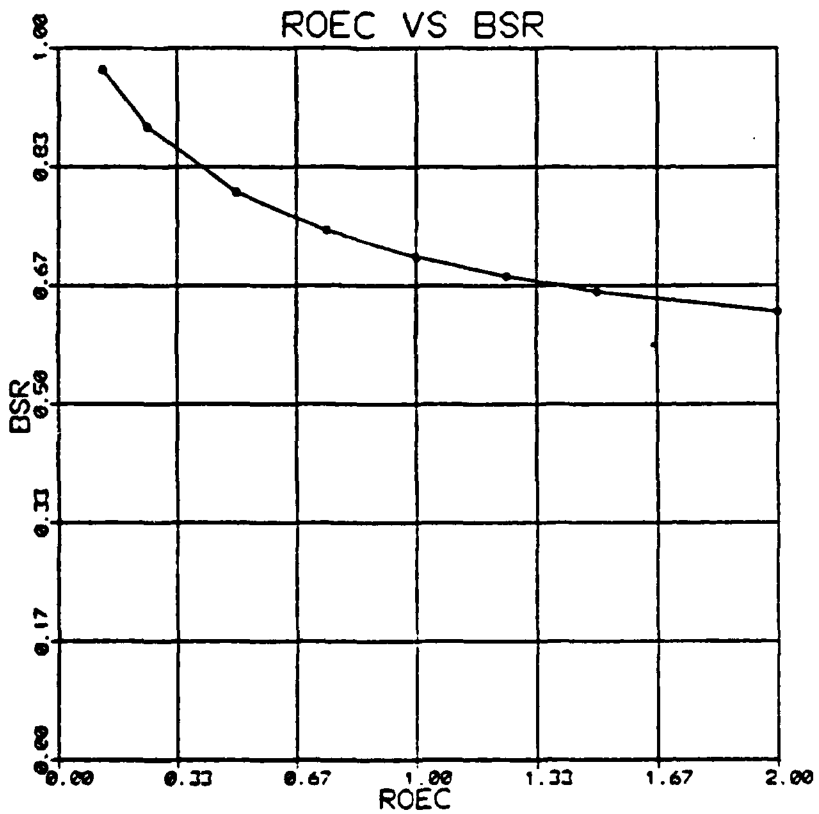

ROEC VS BSR
BSR
1.00
0.83
0.67
0.50
0.33
0.17
0.00
0.00
0.33
0.67
1.00
1.33
1.67
2.00
ROEC

Fig. B

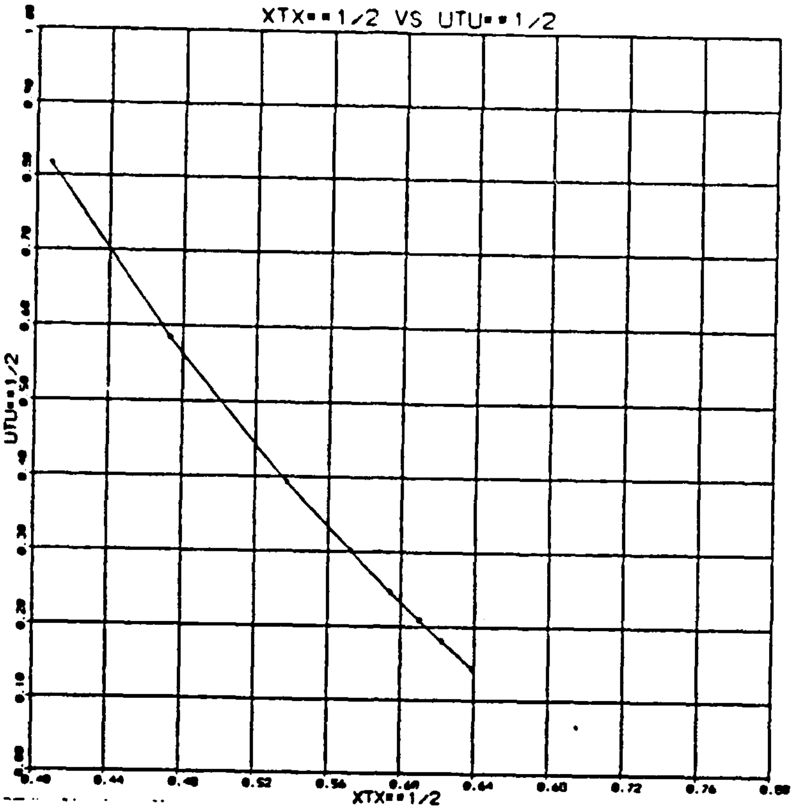

XTX**1/2 VS UTU**1/2
UTU**1/2
XTX**1/2

Fig. C

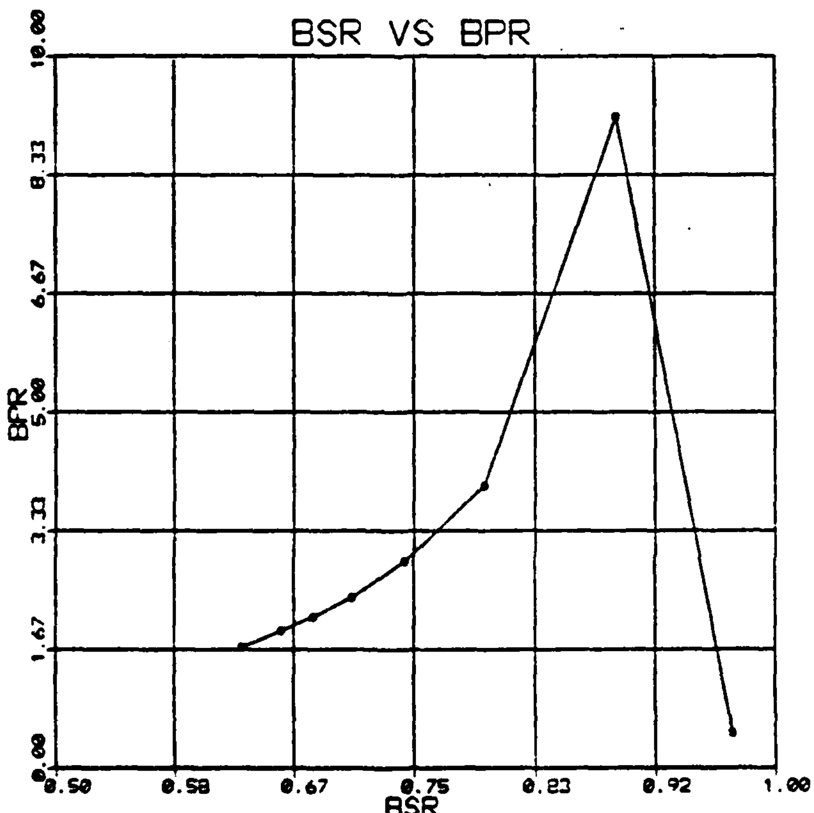

BSR VS BPR
BPR
BSR
10.00
8.33
6.67
5.00
3.33
1.67
0.00
0.50
0.58
0.67
0.75
0.83
0.92
1.00

Fig. D

III. Conclusions and Recommendations for Future Research:

The main theme of the Mini grant research has been to investigate further into the development of a stability robustness condition in time domain and the extension of these results in the computer implementation of a robust control design algorithm that incorporates both stability robustness and performance robustness into the control design procedure. Towards this direction, first a new stability robustness condition is developed in time domain (in terms of eigenvalues) and it is shown that the proposed time domain condition is less conservative than the corresponding frequency domain condition as well as another recently developed time domain condition. Also, further observations are made in the comparison of proposed time domain development to the frequency domain development. Then new measures of 'stability robustness' and 'performance robustness' are developed and incorporated into the robust control design algorithm proposed in the summer research. Finally, computer software is developed to implement the proposed control design algorithm and examples are presented which involve the use of the software.

The experience with Large Space Structure examples carried out indicates that for these models the stability condition in its present form is still conservative and that more research is needed to specialize the analytical development to LSS models.

As it normally occurs, another result of this study is that many interesting research topics surfaced for further investigation. In the next

section, we project those areas which merit serious research effort in the form of shponsorship from AFOSR.

Avenues for Further Research:

1) The foremost area of research would be to look further into the stability robustness condition from two viewpoints.

a) To improve the 'optimism' of the proposed condition particularly with reference to LSS models.

b) To investigate the possibility of developing a new stability robustness condition which is both necessary and sufficient. (Recall that the present condition is a sufficient condition).

2) Another area of immediate concern is to arrive at an algorithm (a technique) that would give the worst case $\beta_{S.R.}$ and worst case $\beta_{P.R.}$ for given perturbations, for comparison purposes.

3) An area of extreme interest would be to develop an algorithm for 'Maximum Allowable Perturbations' that would destabilize a given stable system. In a way this is an 'inverse' problem. This problem is apparently related to the task number one indicated above.

4) An important area of research is to investigate any computation reduction schemes for the proposed algorithm.

5) It is also of interest to probe further into the relationship between frequency domain treatment and the proposed time domain treatment.

IV. References

1. Balas M.J: "Trends in Large Space Structure Control Theory: Fondest Hopes, Wildest Dreams". IEEE Trans. Autom. Control., Vol. AC - 27, #3, June, 82. pp. 522-535.

2. Burns, J. and Cliff E: "Identification and Optimal Control of Aeroelastic Systems", IEEE Conf. on Decision and Control., Dec. 14-16, 1983, San Antonio, Texas.

3. Lehtomaki et. al. "Robustness Tests Utilizing the Structure of Modeling Error". Proceedings of Conference on Decision and Control (CDC) 1981, pp. 1173-1190.

4. Lehtomaki, N.A., Sandell, N.R. and Athans, M., "Robustness Results in Linear Quadratic Guassian Based Multivariable Control Designs", IEEE Trans. on Automatic Control, Vol. AC-26, pp. 75-92, 1981.

5. Lee, W.H., Gully S.W., Eterno, J.S. and Sandell, N.R., Jr., "Structural Information in Robustness Analysis", Proceedings of the American Control Conference (ACC), 1982, pp. 1040-1045.

6. Sandell, N.R., Gully S.W., Lee, W.H. and Lehtomaki, N.A., "Multivariable Stability Margins for Vehicle Flight Control Systems", Proceedings of CDC, pp. 1479-1483., 1981.

7. Mukhopadhyay, V., and Newsom, J.R., "Application of Matrix Singular Value Properties for Evaluating Gain and Phase Margins of Multiloop Systems", Proceedings of the AIAA Guidance and Control Conference, 1982, pp. 420-428.

8. Gupta, N.K., "Frequency Shaped Penalty Functions for Robust Control Design", Proceedings of ACC, 1982, pp. 1035-1039.

9. IEEE Transactions on Automatic Control, Special Issue on Linear Multivariable Control Systems, Vol. AC-26, No. 1, February, 1981.

10. Kosut, R.L., Salzwedel, H., and Emani-Naeini, A., "Robust Control of Flexible Spacecraft"., Journal of Guidance, Control and Dynamics., Vol. 6., #2, March-April 1932. pp. 104-111.

11. Freudenberg, J.S., Looze, D.P. and Cruz, J.B., Jr. "Robustness Analysis using Singular Value Sensitivities", Proceedings of CDC, 1981, pp. 1158-1166.

12. Haddass, Z., and Powell, J.D.: "Design Method for Minimizing Sensitivity to Plant Parameter Variations", AIAA Journal, Vol. 13, Oct. '75, pp. 1295-1303.

13. Ly, V. and Cannon, R.H., Jr., "A Direct Method for Designing Robust Optimal Control Systems", Proceedings of AIAA Guidance and Control Conference, Aug. 1978, pp. 440-448.

14. Yedavalli, R.K. and Skelton, R.E., "Controller Design for Parameter Sensitivity Reduction in Linear Regulators" Journal of Optimal Control: Applications and Methods, by John Wiley & Sons., Vol. 3, #3, July-Sept. 82, pp. 221-240.

15. Yedavalli, R.K., "Time Domain Analysis and Synthesis of Robust Controllers for Large Scale LQG Regulators", Final Report, AFOSR Summer Faculty Research Program, Summer 1982.

16. Kantor, J.C., Andres, R.P., "Characterization of 'Allowable Perturbations' for Robust Stability"., IEEE Transactions on Automatic Control., Vol. AC-28., #1., January, 1983., pp. 107-109.

17. Berman, A., Plemmons, R.J., Non-negative Matrices in the Mathematical Sciences., Academic Press., 1979.

18. Lee, W.H.: "Robustness Analysis for State Space Models" Alphatech Inc., TP-151., Sept. 1982.

19. Kwakernaak H., and Sivan, R., "Linear Optimal Control Systems", Wiley Interscience, 1972.

20. Skelton, R.E., "Cost Decomposition of Linear Systems with Application to Model Reduction"., Int. Journal of Control., Vol. 32., #6., 1980, pp. 1031-1055.

21. Banda S.S., Ridgely, D.B., and Hsi-Han Yeh, "Robustness of Reduced Order Control"., Presented at the VPI&SU/AIAA Symposium on Dynamics of Control of Large Structures., Blacksburg., Va., June 6-8, 1983.

22. Yeh, H.H., Banda, S.S., and Ridgely, D.B., "Stability Robustness Measures Utilizing Structural Information", American Control Conference, June 6-8, 1984, San Diego and also submitted for publication in the IEEE Transactions on Automatic Control.

23. Yeh, H.H., Ridgely, D.B., Banda, S.S., "Nonconservative Evaluation of Uniform Stability Margins of Multivariable Feedback Systems", AIAA Guidance and Control Conference, Aug. 20-22, 1984, Seattle, Washington.

24. Lancaster, P., "Theory of Matrices", Academic Press, 1969.

25. Barnett, S. & Storey, C., "Matrix Methods in Stability Theory" Barnes & Noble, Inc. N.Y., 1970.

26. Korn, G.A. & Korn, T.M., "Mathematical Handbook for Scientists and Engineers"., McGraw Hill: 2nd ed., 1968.

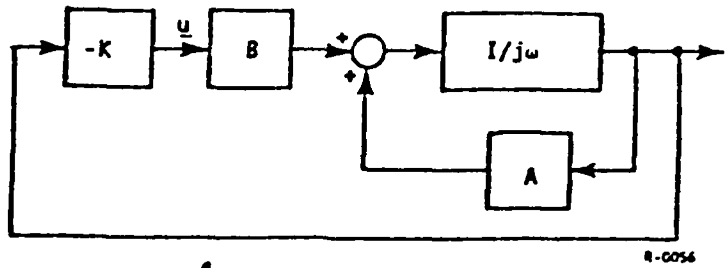


Fig. 1 . State Feedback Regulator

Appendix A

Proof of Theorem 1:

Let $\rho\{[E_s(F_s)^{-1}]_s\}<1$

$\rightarrow \quad |\lambda(E_s(F_s)^{-1})_s|_{max} < 1$

$\rightarrow \quad |\lambda_i(E_s(F_s)^{-1})_s| < 1 \quad \forall i$

$\rightarrow \quad 1 + \lambda_i\{[E_s(F_s)^{-1}]_s\} > 0 \quad \forall i$

$\rightarrow \quad \lambda_i\{I + (E_s(F_s)^{-1})_s\} > 0 \quad \forall i$

$\rightarrow \quad \lambda_i\{[I + E_s(F_s)^{-1}]_s\} > 0 \quad \forall i$

$\rightarrow \quad [I + E_s(F_s)^{-1}]$ is positive definite

$\rightarrow \quad [I + E_s(F_s)^{-1}]\,[-F_s]$ has positive, real eigenvalues because 1) if A and B are positive definite, AB has positive real eigenvalues. (Ref [24,25])

and

2) If A is negative definite, -A is positive definite and hence $-F_s$ is positive definite (Ref [26]).

$\rightarrow \quad -(F_s + E_s)$ has positive, real eigenvalues
[because $[I + E_s(F_s)^{-1}]\,[-F_s] = -(F_s + E_s)$]

$\rightarrow \quad -(F_s + E_s)$ is positive definite (because $-(F_s + E_s)$ is symmetric too)

$\rightarrow \quad (F_s + E_s)$ is negative definite

$\rightarrow \quad (F+E)_s$ is negative definite

$\rightarrow \quad (F+E)^s$ is negative definite

$\rightarrow \quad (F+E)$ has negative real part eigenvalues

$\rightarrow \quad (F+E)$ is stable .

END

FILMED

3-84

DTIC